

---

# Hands-on

## Integration Competency Center

---

Handbook

Version 4.0

3 November 2015

Antti Toivanen, Katja Kuusela



[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 3.0 Unported License](#)

<b>1</b>	<b>FIGURES</b>	<b>4</b>
<b>2</b>	<b>DEFINITION</b>	<b>5</b>
2.1	Why ICC	5
2.2	Benefits in numbers	6
<b>3</b>	<b>INTRODUCTION</b>	<b>8</b>
3.1	Establishing ICC	9
3.2	Requirements	10
3.3	Common Challenges	11
<b>4</b>	<b>ICC AND ENTERPRISE ECOSYSTEM</b>	<b>13</b>
4.1	<b>Roles</b>	<b>14</b>
4.1.1	ICC Director	15
4.1.2	ICC Manager	17
4.1.3	Business Information Specialist	19
4.1.4	Integration Architect	21
4.1.5	Functional Designer / Information Architect	23
4.1.6	Technical Designer	24
4.1.7	Service Manager	26
4.1.8	Quality Manager	26
4.1.9	Security Architect	28
<b>5</b>	<b>DELIVERY MODELS</b>	<b>31</b>
5.1	Artifact Driven Model	31
5.2	Scrum model	31
5.2.1	Scrum roles in ICC	32
5.3	Kanban Model	33
<b>6</b>	<b>ICC ACTIVITIES</b>	<b>36</b>
6.1	<b>Recurring Events</b>	<b>36</b>
6.1.1	ICC Meetings	36
6.1.2	Steering Group Meeting	39
6.1.3	Lean Session	39
6.1.4	Disaster Recovery Test	40
6.2	Communication Gap	40
<b>7</b>	<b>BEST PRACTICES</b>	<b>43</b>

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

<b>7.1</b>	<b>Eliminate Waste</b>	<b>43</b>
<b>7.2</b>	<b>Service Oriented Architecture</b>	<b>44</b>
<b>7.3</b>	<b>SOA and Microservices</b>	<b>46</b>
<b>7.4</b>	<b>Cyclic Deployment</b>	<b>47</b>
<b>7.5</b>	<b>Testing</b>	<b>51</b>
<b>7.6</b>	<b>Test-driven development</b>	<b>52</b>
7.6.1	Acceptance Test-Driven Development	53
<b>7.7</b>	<b>Artifacts Review</b>	<b>54</b>
<b>7.8</b>	<b>Canonical Enterprise Data</b>	<b>55</b>
<b>8</b>	<b>CONCLUSION</b>	<b>57</b>
<b>9</b>	<b>APPENDIX - A: ICC CHECKLIST</b>	<b>59</b>
<b>10</b>	<b>APPENDIX - A: PERSONS ASSIGNED TO ICC ROLES</b>	<b>63</b>
<b>11</b>	<b>REFERENCES</b>	<b>64</b>
<b>12</b>	<b>BIBLIOGRAPHY</b>	<b>65</b>

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

# 1 Figures

<i>Figure 1: ICC Benefits</i> .....	7
<i>Figure 2: ICC stages and associated activities</i> .....	10
<i>Figure 3: ICC and Enterprise Ecosystem</i> .....	13
<i>Figure 4: ICC with built-in integration factory and platform management</i> .....	14
<i>Figure 5: ICC Roles</i> .....	15
<i>Figure 6: Scrum model with stints</i> .....	32
<i>Figure 7 Sample Kanban table</i> .....	34
<i>Figure 8: Activities Flow</i> .....	36
<i>Figure 9: Communication flow</i> .....	42
<i>Figure 10: Continuous Development process</i> .....	48
<i>Figure 11: Sequencing problem</i> .....	49
<i>Figure 12: Development and Deployment Cycle</i> .....	51
<i>Figure 13: Test Driven Development</i> .....	52
<i>Figure 14: Sample O2C process</i> .....	55

[Back to Contents](#)

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

## 2 Definition

A **competency** is an underlying characteristic of an individual that is causally related to a certain criterion which is referenced as *effective* and *superior performance* in a job or other professional activities. (Spencer & Spencer 1993: 9)

There are two perspectives: The **work-oriented competences** which are regarded as a specific set of attributes of work and its demands, whereas the **worker-oriented competences** are primarily seen as constituting attributes possessed by a professional (an individual), typically represented as knowledge, skills, abilities and personal traits required for effective work performance. Thus, **Competency** refers to the *expertise, knowledge* or *capability* that are offered as services.

And a **Center** means that, the service is managed or coordinated from a common (central) point independent from the functional areas that it supports. (Wikipedia, 2012)

Therefore, a **Competency Center** can be associated to a composition of varying competencies necessary for an execution of complexly related tasks to achieve a wholesome goal.

**Integration** as applied in **ICC**, refers to system integration which connect business systems together at data level and Business Process Automation which automate the business processes running on top of business applications like ERPs and CRMs.

And hence, we can define **ICC-Integration Competency Center** as an organization composed by a group of professionals with differing competencies whose purpose is to execute different and complex tasks with the aim of achieving a complete integration in a given enterprise environment.

### 2.1 Why ICC

There are four typical problems in most of organizations (big or small):

1. Communication - Even when there are well defined processes in place, communication is most of the times a key problem. What we mean by communication is the parties which should be involved, the flow/direction of communication and the content of what is being communicated at a given time.
2. Web of processes - This is typical in big organizations. Thicker processes leads to bureaucracy; and bureaucracy can be a dent in doing business especially within an organization and with organization's partners.
3. Process absence - In some organization, a critical business activity might develop without proper process in place - an absolute explosive situation, sooner or later.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

4. Quantum and Pace of changes – Speed is a key factor for business development; from integration point of view this leads to large amount of changes that has to be in production fast. As errors in integration solution are usually a high expense, speed cannot be made achieved at the cost of quality.

How does ICC answer to that?

1. Typically, ICC is a shared Service within an organization. However, it should be understood that, by being central, does not mean ICC should centralize all processes.
2. What ICC does is to follow, develop, facilitate and enhance organization's processes with the aim to make them lighter (agile), efficient (non-wasteful) and relevant (non-bureaucratic).
3. Data Integration: Allows companies to access their **enterprise data** and functions, fragmented across disparate systems, in order to create a *combined, accurate* and *consistent* view of their **core information** as well as process assets and leverage them across the entire enterprise to drive business decisions and operations. (Wikipedia, 2012)
4. System Integration: Is the bringing together of components subsystems into one system and ensuring that they function together effectively. (Wikipedia, 2012)
5. Process Integration: Enables efficient information exchanges and business process automation across separate applications within (EAI) and across (B2Bi) organizations in a unified fashion.
6. ICC involves all stakeholders such as business people and main users in active communication, continuous process development and glues development and operating together.
7. ICC defines Best Practices and ensures that they are really taken into use in daily development and operating tasks

**Integration Competency Center** - ICC provides the following key benefits:

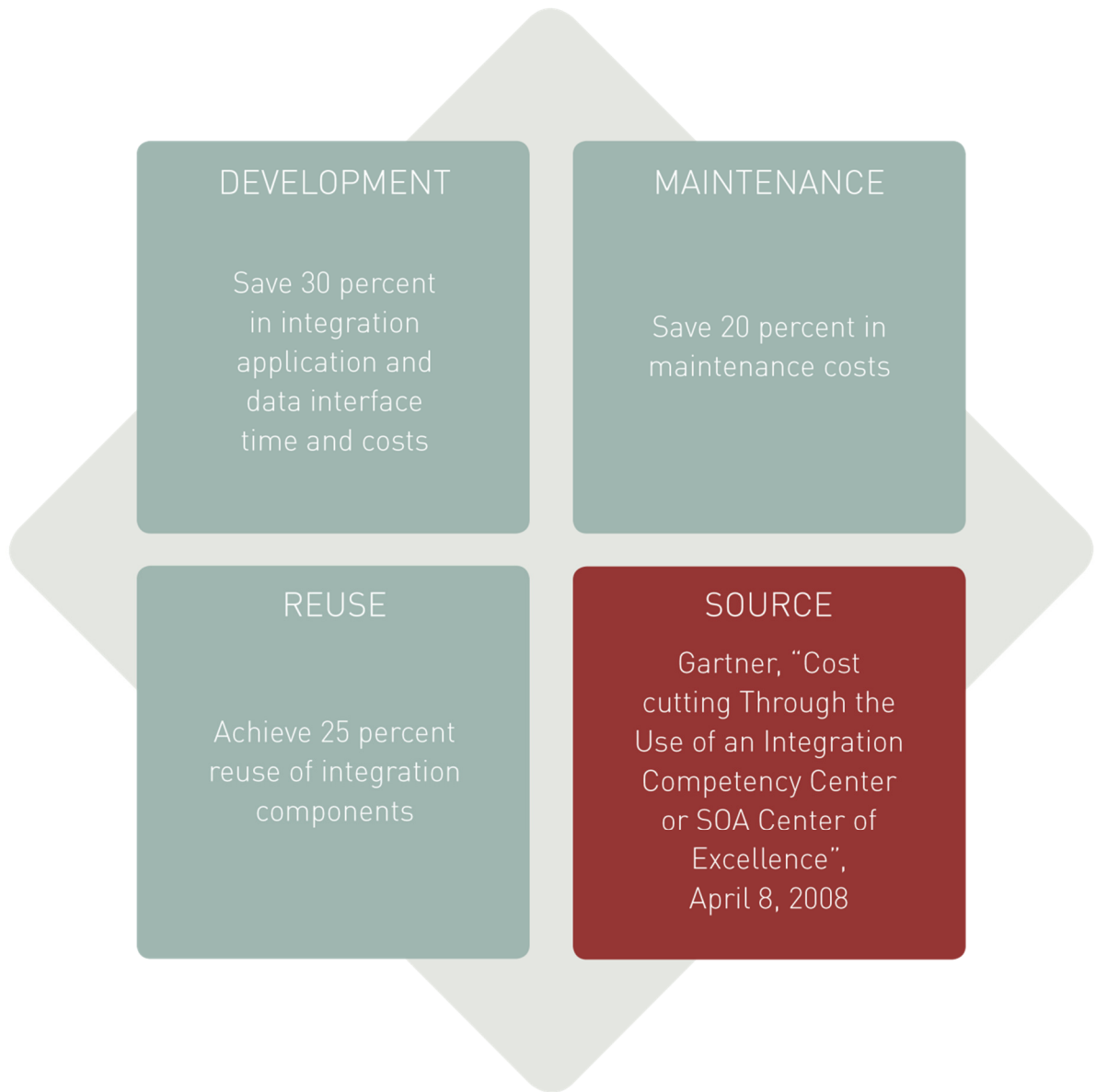
1. Quality through controlled processes executed by competent personnel (roles) using best practices
2. Cost-savings through reusability
3. Easier maintenance through reusability and quality
4. Agility to keep up with the increasing pace of changes.
5. Helps to gain overall view of a company's processes, systems and technologies
6. Defines the technical representation of business processes and IT infrastructure

## 2.2 Benefits in numbers

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)



**Figure 1: ICC Benefits**

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

### 3 Introduction

With 20 years of experience in business process integration, HiQ-ICC has a unique understanding of something that has been commonly known as [Integration Competency Center](#) - ICC. You can easily find studies which tell what to do, but not how to do it. However, this document outlines key aspects that HiQ-ICC has noted as vital elements for successful integration as a result of years of experience in the field of integration. To start with let us outline the main ICC elements.

1. Organization and Roles
2. Responsibilities of the roles
3. Integration Best Practices
4. Processes
  - A. Formalized steps for each process action
  - B. Document trace for each process
  - C. Continuous elimination of waste in processes
5. Standardization
  - A. Naming conventions
  - B. Common standards
  - C. Process standardization

ICC should not be confused with a project or process model, instead ICC is a shared service intended for companies who wish to design, develop and maintain integration solutions in an integration platform serving more than one Business Unit or Project.

ICC's responsibility is to define and choose the roles and processes i.e. Incident Management, Problem Solving, Project Delivery, etc. Additionally, for successful development and maintenance of an integration solution, the following best practices or activities are mandatory:

- Cyclic Deployment
- Detailed testing procedures, preferably automated
- Practically tested Disaster Recovery
- Active communication
- Transition process
- Requirement management

These best practices are described later in [Chapter 6](#).

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

## 3.1 Establishing ICC

This chapter covers basic steps to establish ICC Organization. Remember that one of the most common reasons for failure in [ICC work](#) is to go straight into “*execution-phase*” without proper ICC establishment.

This approach has four main requirements:

1. Establish the ICC organization
  - A. Find out the persons to fill out the roles defined in [chapter 3](#).
  - B. Use internal and external resources
  - C. Find ICC Manager soon enough to run the establishment phase
  - D. Ensure funding (resources etc.)
  - E. Communicate benefits of ICC to stakeholders
  
2. Define or choose the processes
  - A. Why go and reinvent the wheel? Are you sure that there is no existing Change Management or Deployment process suitable for your organization?
  - B. Train and communicate processes to ICC group and other stakeholders
  
3. Establish communication procedures
  - A. Define strict agenda for weekly, monthly and other meetings
  - B. Weekly meetings should follow-up the ICC’s establishment phase tasks in the beginning
  - C. Train and communicate procedures to ICC group and other stakeholders
  
4. Establish Systems
  - A. Version control
  - B. Test systems
  - C. Centralized Document Management
  - D. Incident Management
  - E. Business Backlog System
  - F. Technical Backlog System
  - G. Service catalog for reusable business services (SOA)

An [ICC](#) is a permanent part of the organization. ICC can be established at various scales or levels – within a division of a company, at the enterprise level or across multiple companies. ICC should scale with the business needs of the company.

In the [Appendix-A](#) you will find a checklist which helps the organization ensuring that everything is in place in order to support ICC establishment process.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

The following steps and associated activities are required when establishing an ICC Service group in an organization.

1. Depending on Customer's processes, we may need to establish ICC from ground-up or adopt the ICC program.
2. Tailor the delivery strategy to the readiness and needs of a specific Organization/Customer
3. Determine according to organizational needs; how ICC should be deployed
4. Adjust the ICC deployment in terms of length and the order of deployment

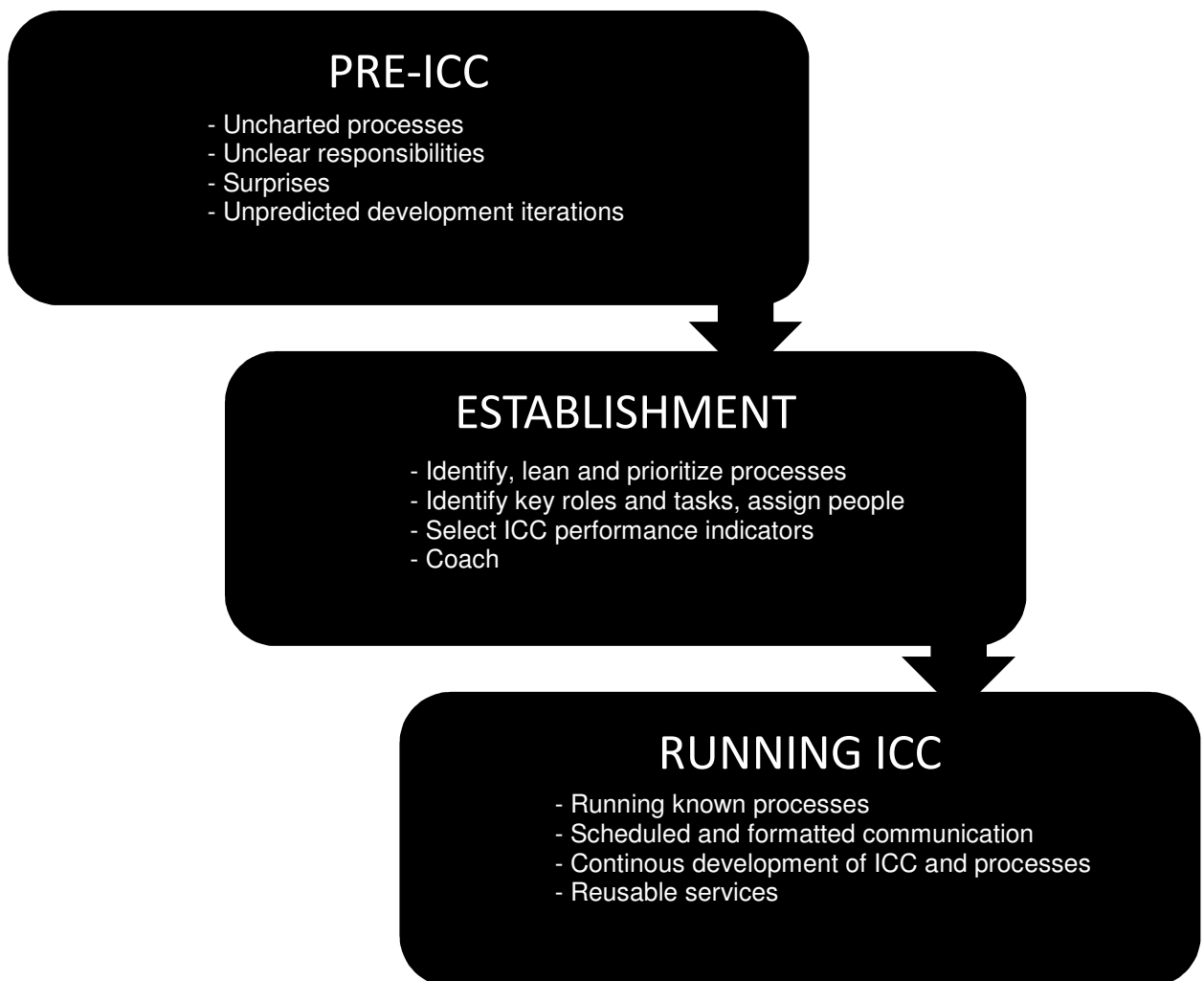


Figure 2: ICC stages and associated activities

## 3.2 Requirements

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

Following elements are mandatory building blocks and procedures for successful [ICC work](#):

1. [ICC roles](#) with competent personnel
2. Well defined processes and system environments
  - A. Actions and persons responsible for those actions are clearly assigned
  - B. Technical environments (integration platforms)
    - Requirement: The combination of Development, Testing and Production is a minimum requirement for environments
    - Recommended: The combination of Development, Testing, Acceptance Testing and Production
3. Formally defined testing procedures. Make sure that the Testing is covering all programmed functions, statements, normal conditions, error conditions and decisions in the workflow.
4. [Cyclic Deployment](#)
5. [Disaster Recovery](#)
  - A. Disaster Recovery without testing it, is not a Disaster Recovery, it is a Lucky-Recovery.
6. Weekly and Monthly meetings
7. Active communication with all stakeholders

### 3.3 Common Challenges

[ICC](#) as a concept is fairly simple. It is embodiment of the IT management best practices to deliver shared services. However, as each organization has a different DNA which brings a specific customization effort to make the ICC initiative successful for that particular organization – it is therefore far more challenging to implement in practice.

The following are some of the most common challenges in ICC establishment journey:

1. Change Management in terms of technology, processes, organization structure.
2. Ability of the organization to deal with the pace and quantum of change.
3. Alignment of stakeholders and process owners for ICC strategy.
4. Inappropriate ownership level for ICC program and lack of senior management sponsorship.
5. Highly tactical focus.
6. Business program level constraints.
7. Ignoring the fundamental elements and jumping to implementation directly.
8. Inappropriate funding.
9. Assumption as a recipe for a disaster.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

“We didn’t touch this part of the Solution, therefore it has to work”

This assumption may lead to a catastrophe. Make your decisions based on facts instead of assumptions: for example, Change management matrix shows that which changes have an impact to systems A and B whereas system C is not affected. Change management matrix indicates which system and solution parts should be regression tested.

#### 10. Cyclic Deployment

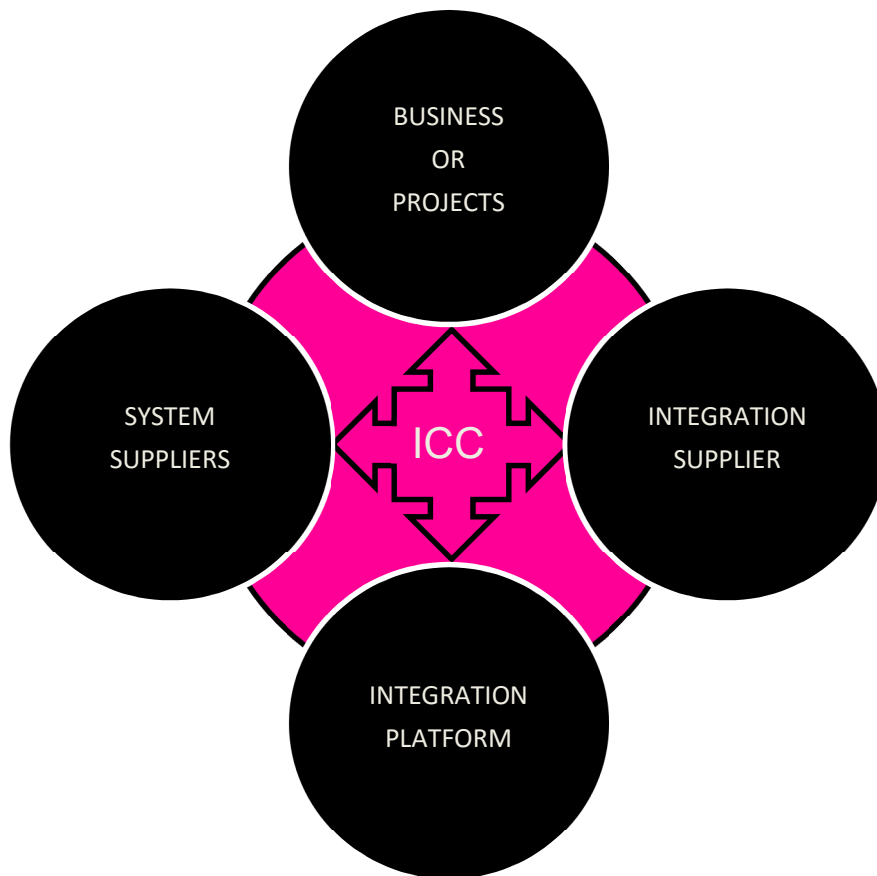
It is good to remember, that the conceptual definition of ICC which is not properly implemented in the organization has no real value for the enterprise.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

## 4 ICC and Enterprise Ecosystem



**Figure 3: ICC and Enterprise Ecosystem**

Integration work includes always several parties. The most common setup is described in Figure above. The picture above illustrates the role of **ICC** – it is a shared service that serves Business Units or Projects or both. ICC manages all **communication** and work assignments between internal business or projects and external System Suppliers, System Integrators and other parties who participate in integration. Managing acceptance and *deployment testing* and *deployment* itself is also a major responsibility of ICC.

As ICC is adjusted to the needs of the implementing company, setup may vary. Other common setup is illustrated below. In this model ICC also implements the integration solutions and manage integration platform directly without separate IT organization.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

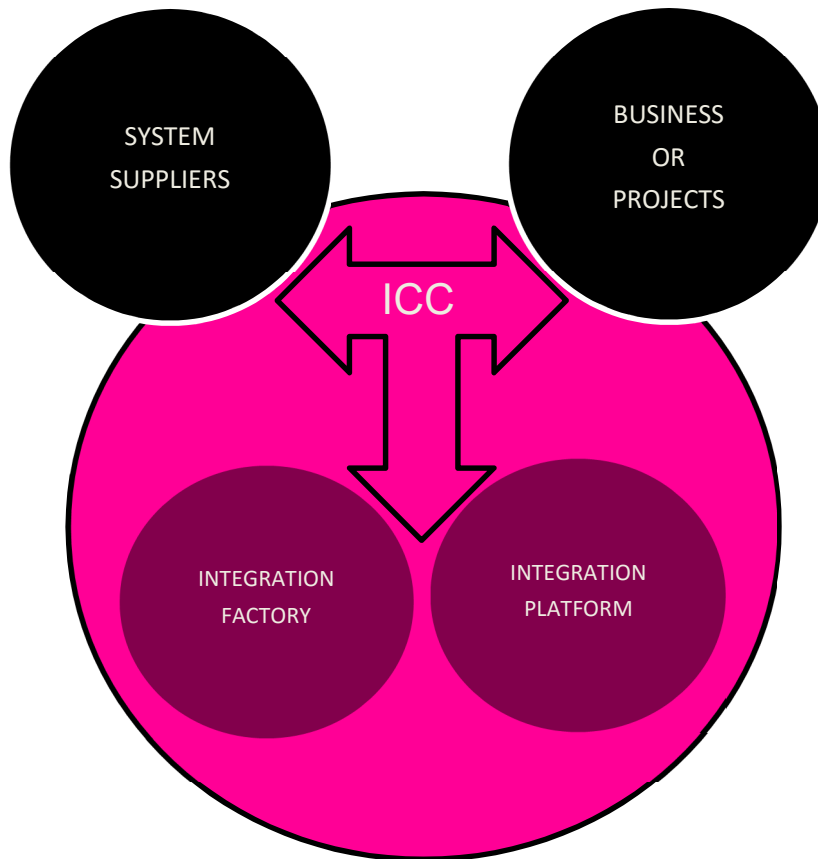


Figure 4: ICC with built-in integration factory and platform management

## 4.1 Roles

This chapter defines tasks and roles what are mandatory or at least beneficial in successful integration management.

One critical issue to remember is to treat the extended team members as member of first-order – this includes incentives, bonuses, team celebrations and individual recognition and rewards.

It must be pointed out that, the need and size of ICC Group will vary with respect to the size of the company.

For big companies, ICC is almost a must, whereas for smaller companies, the ICC Group can be sized down to a one man band.

That said, in smaller companies, where ICC is ran as a "one man band", there will certainly be benefits in subcontracting critical expertise to fill some of the crucial roles in the ICC Group.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

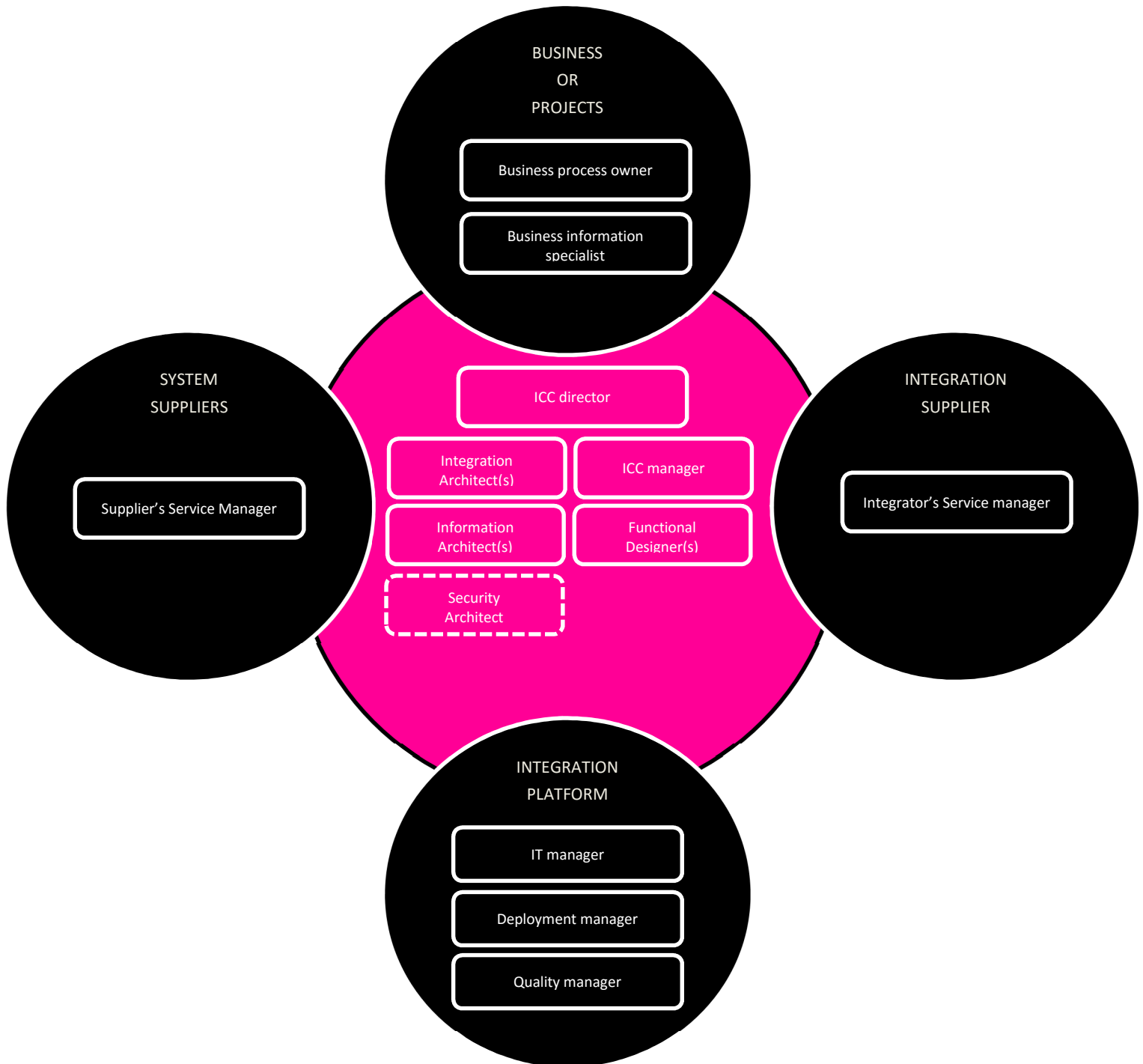


Figure 5: ICC Roles

### 4.1.1 ICC Director

**Reports to:** Business Executive.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

ICC Director has overall responsibility for planning and managing an [Integration Competency Center](#). Strong leadership of ICC Director is especially important in establishment phase of [ICC Organization](#).

Specific activities vary depending on

- ICC scope
- ICC maturity level
- Organizational model

The ICC Director works with

- Project teams
- Technology groups
- Business Architects
- Cross-functional management teams, for example ERP development team.
- Third parties technology suppliers.
- Other stakeholders to plan and operate the ICC

Specific responsibilities:

- Project Model.
- Problem Solving.
- Disaster Recovery.
- Production Deployment.
- Continuous Development.
- Establish required systems.
  - Version Control
  - Incident Management System
- Establish a 30/60/90/120+-day plan.
- Select processes to be supported by ICC.
  - Project Model
  - Problem Solving
  - Disaster Recovery
  - Incident Management
  - Production Deployment
  - Continuous Development
  - Transition (from development to production)
  - Requirement management
- Resolve prioritization and budget issues.
- Establish required systems and applications.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

- Manage [ICC processes](#), operating rhythm and budget.
- Develop and manage the [ICC work](#) plan and service levels.
- Define metrics and values to monitor and measure projects.
- Communicate progress and exceptions to project sponsor(s).
- Ensure delivery on commitments and service level fulfilment.
- Determine the scope and mission of the ICC and gain executive support.
- Define the organizational model for the ICC and its core operating principles.
- Work with the management teams to procure and assign the appropriate resources.
- Establish [ICC Organization](#) by selecting competent people to the crucial [ICC roles](#).
- Define the Integration Strategy and communicate its progress to the Steering/Management Group.
- Selects vendors for the integration platform and development.

#### 4.1.2 ICC Manager

**Reports to:** [ICC Director](#). May also report to a C-level executive such as CIO, CTO, or COO.

ICC Manager serves as the right hand of ICC Director.

**NOTE:** This role is mandatory.

The ICC Manager has overall responsibility for planning and leading daily [ICC work](#).

ICC Manager is usually hired as an in-house employee, but in some special arrangement ICC Manager can be a consultant and therefore work on behalf of the customer/client.

ICC Manager can be compared to Project Manager; he or she must keep everything under control and make sure that overall solution is developing uninterrupted.

ICC Manager serves as the right hand of ICC Director running the daily and weekly tasks.

Specific activities vary depending on

- Project scope
- Project life-cycle
- Process maturity level
- Organizational model

The ICC Manager works with

- ICC Director
- Project teams

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

- Technology groups
- Business Architects
- Cross-functional management teams
- Technology suppliers i.e. third parties
- Other stakeholders to plan and operate the ICC projects

Main activities are

- Overseeing the whole project
- Preparing project update reports
- Organizing weekly / monthly meetings
- Prioritization and scheduling of resources
- Follow-up business backlog items e.g. INTs
- Drafting of reports for senior management
- Decide which document each implementation should produce independent of project or process model
- Any other management task

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

**Responsibilities:**

- Coordinates training of best practices, including but not limited to naming conventions, unit test plans, configuration management strategy, and project methodology.
- Provides full-time management resources experienced in data integration to ensure project success.
- Identification and reporting of potential project delays on an on-going basis.
- Preparing project schedules, ensuring co-ordination with other affected parties to avoid scheduling conflicts.
- Preparing and presenting detailed schedules in any agreed format, with clearly distinguishable milestones and if possible/required providing a critical path.
- Providing project management oversight and co-ordination from the client perspective for projects administered by third parties.

**Practice:**

- Among other tools which can be used in managing integration projects under ICC, one main philosophy which could be of use to know is – the [Deming's 14 principles](#).

### 4.1.3 Business Information Specialist

**Reports to:** Business Unit Manager.

Business Information Specialist is a representative of Business Unit or Project. She or he is responsible of identifying the need for integration and initializing the defining of requirement for ICC.

**NOTE:** Business Information Specialist role is optional. It is quite common that Business information specialis tasks are done by ICC's Information architect.

Specific activities vary depending on

- Industry sector
- Business model

The Business Information Specialist works with

- [Functional Designer](#)
- Business Architects
- Cross-functional management teams

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

Main activities are

- Write the initial requirement for ICC
- Provide understanding of information
- Support requirements definition and capture business activities

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

**Responsibilities:**

- Maintain the information on Master Data
  - Knowledge of main systems i.e. ERP, CRM and SCM
  - Knowledge of main systems Master Data
  - Knowledge of unique key values of Master Data
  - Maintain and publish documentation on how the Master Data is published from the main systems into integration layer. For example maintain a registry on existing Web Services.
- If enterprise canonized data model (see best practices) is used, canonize data for ICC and enterprise use (if not already done)
- Evaluate reusability of the information that is published via Web Service
- Find out the Master Data for requirement from the main systems i.e. ERP, CRM and SCM
- Evaluate requirement against existing information
  - Can requirement be fulfilled with available information
- Evaluate whether the main systems can actually perform the required operations as specified in the requirements. E.g. low-latency requirement
- Develop and maintain Business Information Modeling Guiding Principles.
- Identify business processes and Service Oriented Architecture (SOA) services that support key business functions. Provide industry insights on information standards and interoperability.

#### 4.1.4 Integration Architect

**Reports to:** [ICC Manager](#). May also report to the [ICC Director](#) and in a special event to a C-level executive such as CIO, CTO, or COO.

**NOTE:** Mandatory in larger integration environments.

Integration Architect is responsible of maintenance the overall integration architecture.

Specific activities vary depending on

- Industry sector
- Company size

The Integration Architect works with

- ICC Manager
- [Business Information Specialist\(s\)](#)

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

- Technology groups
- Cross-functional management teams
- Technology suppliers i.e. third parties

Main activities are

- Defining and maintaining integration Patterns i.e.
  - General rules for integration
  - Integration patterns
  - SOA and ESB implementation and governance
  - Service awareness and internal-marketing
- Develop the architecture in a given sector
- Collaborate with other architectural groups
- Maintain Server topology documentation
- Provide technical insights of the integration solution
- High Availability and Scalability planning with respect to requirements
- Policing that every solution do follow the set integration rules
- Helps the Functional Designer to translate business requirements into technical solutions
- Ensuring that all implementation decisions do follow the integration strategy
- Retain ownership of architectural issues prior to project sign-off

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

**Responsibilities:**

- Choosing the standards and recommendations i.e. messages structure and sequence as well as processes to follow in integration tasks.
- Assisting during the pre-sales analysis, scoping and designing phases.
- Verifying that the pre-sales scoping, analysis and design deliverables are technically feasible and conducting system prototyping and testing as appropriate.

#### 4.1.5 Functional Designer / Information Architect

**Reports to:** [ICC Manager](#) and/or [ICC Director](#). May also report to a C-level executive such as the CIO, CTO, or COO.

**NOTE (1):** This role is mandatory. This role is often carried by same person who is on the [Business Information Specialist](#) role.

**NOTE (2):** For specifying data integration – Functional Designer designs the integration process and gathers artifacts for implementer. For specifying process integration – a [Technical Designer](#) is always needed.

Functional Designer acts as a link between the “business need” and the technical solution. She or he should analyze the “business need” and find the artifacts required by Technical Designer and/or Implementer. If business information is canonized during the development of e.g. SOA creation, this role is often called Information Architect.

##### The Functional Designer

- Specifies “what is happening” in all Use-Cases.
- Is responsible with the initial creation of all Use-cases/stories/features with their elements i.e. Actors, Scenarios, Successful paths and Exceptions.
- Points out “what are the expected “Exceptions””.
- Defines “what should be done” if an exception occurs.

The Functional Designer can always ask for support from the people responsible with the following roles:

- Integration Architect – Choosing the best design pattern for the solution.
- Business Information Specialist – Questions concerning the Master-Data, Data mapping or other artifacts.
- ICC Manager – In conflict situations

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

Main activities are

- Refine the Functional Requirements in “business backlog”
- Add following information to “business backlog” items
- Oversee Use Cases
  - Extract and dig out data-mapping rules
  - Flowchart from functional point of view
  - Find out any special rules like extra validation, data enrichment from third system etc.
  - Messaging sequence picture (e.g. by UML) - if the functionality requires more than one request-response pair
  - Design Exception handling
    - Find out which exceptions may occur at functional level. For example a cancellation message for an existing Work Order is rejected by ERP because the work has already been started.
    - Specify what happens if one of the systems that are being integrated does not answer. For example, a communication failure is a technical issue; we should prepare the logic of integration process to handle the exception.
- Gather and document Integration Test Cases
  - Based on the Use Cases and Functional Requirements
  - There should ALWAYS be more than one Test Case testing successful execution path and also always more than one Test Case testing unsuccessful execution path
- Arrange meeting with business people to
  - Find out answers to any open functionality related questions that were raised during functional design, technical design or implementation
  - Make sure that all artifacts are available
- Retain ownership of functional issues prior to project sign-off

Responsibilities:

- Working with Quality Assurance team in producing development Test-cases and test-plan.
- Translate business requirements into technical solutions.

## 4.1.6 Technical Designer

**Reports to:** [ICC Manager](#) and/or [ICC Director](#). May also report to a C-level executive such as the CIO, CTO, or COO.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

**NOTE (1):** In agile models, Technical Designer is most likely the same person as Implementer. Furthermore, in smaller ICC Organizations the [Integration Architect](#) and Technical Designer are very likely to be carried out by the same person.

**NOTE (2):** If the requirement is a simple Data Flow with Data Transformation without complexity, it is possible that the technical specification will not be needed. Also, in a simple integration cases, Functional Specifications with general rules like naming conventions, error handling and logging are enough for Implementer.

**NOTE (3):** This role is mandatory in technically complex solutions. Complex integration case can include messaging scenarios with more than one request-response pairs that update stateful object in a persistent storage like Database or an ERP System.

NOTE (4):

- The Technical Designer will have to point out any missing “Exceptions” from the technical point of view.
- The Technical Designer will make sure that all process steps in all Use-Cases are technically possible.
- The Technical Designer is always needed if a Process Integration is a requirement.

Technical Designer must have the ability a solution for the complex requirement. In addition, he or she writes a technical specification of how requirements are met including non-functional requirements. These may include for example requirement for low-latency in Web services due to User-interface usability.

In an agile integration delivery model, implementation can start with detailed Functional Specification and Non-Functional Requirements

After implementation, a proper technical documentation on how requirements were met must be manually written or automatically generated by specific auto-documenter.

Main activities are

- Write the Technical Specification if needed.
- Transform the Use Cases defined by Functional Designer to Technical Specifications and eventually as “technical backlog”.
- Requirements should be split into atomic amount of work that is easily assignable to single person for implementation.
- Retain ownership of technical support issues prior to project sign-off.

Responsibilities:

- Write Technical Specification when required.
- Specify complex processes.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

- Specify how to utilize more complex technologies.
- Design and develop new enhancements and features for Web Application.
- Perform code review and mentor developers in code implementation to follow coding standard and best practices.

#### 4.1.7 Service Manager

**Reports to:** [ICC Manager](#) and/or [ICC Director](#). Service Manager may also report to a C-level executive such as the CIO, CTO, or COO.

**NOTE (1):** Depending on chosen project model this role is called Service Manager for [Continuous Development](#) or Project Manager for project methodologies.

**NOTE (2):** If ICC manages smaller changes with Continuous Development model and larger development work with Project model (i.e. AIM) then in smaller ICC Organizations it is common that one person manages both of these models: Continuous Development and the Project itself.

Main activities are

- Managing contents of each *deployment package*.
- Running the chosen Project or/and *development model*.
- Reporting to ICC Manager about all development activities.
- Managing ICC Customer account at Integration Service Provider side.

Responsibilities:

- Costs
- Schedule
- Project management at Integration Service Provider

#### 4.1.8 Quality Manager

**Reports to:** [ICC Manager](#) and/or [ICC Director](#). May also report to a C-level executive such as the CIO, CTO, or COO.

**NOTE:** In simpler cases without business critical solutions, this role is not always required.

Quality Manager is responsible for overall quality of deployed integration solutions.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

This role is commonly set to ICC Manager. In larger integration installations reconsider this setup. Remember that understanding the testing of a complex integration solution is exceptionally challenging. Therefore, we suggest to carefully select a competent professional for this role. A good choice is to use 3rd party who focuses on testing and quality assurance.

Main activities are

- Ensuring that “Integration Tests” schedules are done on time.
- Ensuring that the right competences are present in each “Integration Test” session.
- Managing, implementing, maintaining and improving of test processes.
- Managing “Integration Test” sessions
  - Ensure that persons with proper competencies are present at Integration Test session.
  - Ensure that test environment is ready before test session begins.
    - All *deployment packages* that were planned to be tested are actually deployed to test environment.
    - Test environment is technically available and user rights are assigned to test personnel.
    - Make sure that other tests that might have impact on the integration tests are not executed concurrently. For example, main ERP stress tests typically have an impact to integration layer.
  - Ensure that Production Deployment documentation is updated accordingly to test results.
  - Ensure that test logs are updated.
- Quality Management
  - Define how Quality is measured
  - Follow-up that quality is really measured
  - Define requirements for solution quality
- Gathering and maintaining measures and metrics of testing.
- Test result reports
  - Review test logs and take necessary actions
  - Keep test documentation up-to-date

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

- Arrange frequent [Disaster Recovery Test](#)
  - It is recommended that Disaster Recovery MUST be tested at least once in every SIX months.

#### Responsibilities:

- Making sure that the Testing team understands role and importance of testing.
- Assuring that other stakeholders understand the meaning of test requirements.
- Making and policing test-discipline i.e. Test Case reviews; are we covering everything with current Test Cases?
- Making sure that testing has appropriate role in overall *development process*.
- Making sure that testing is considered early on designing new solutions are designed or existing ones are changed.
- Making sure that needs for time and other resources are communicated clearly.

### 4.1.9 Security Architect

**Reports to:** [ICC Director](#). May also report to a C-level executive such as the CIO, CTO, or COO.

**NOTE:** The Security Architect must ensure that the information is secured but yet accessible to those processes that consume the information.

Plans, executes and audits security issues in all ICC projects. And makes sure that security issues are covered in processes and deliverables.

The Security Architect works with

- ICC Director and [ICC Manager](#)
- Project teams
- Technology groups
- Business Architects
- Cross-functional management teams
- Technology suppliers i.e. third parties
- Other stakeholders

Main activities are

- Security reviews.
- Security objectives.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

- (Role based) authorization.
- Lead security incident management activities.
- Access control to the Integration solution and its information.
- Drive key architecture in the areas of overall security and access control.
- Communicate with various collaborators and third parties.
- Assist in planning and testing the security of managed services.
- Consult service delivery teams on security considerations, best practices, and patterns.
- Architect, design and assist on implementation of security-related systems and functionality.
- Assist in and potentially conduct internal vulnerability assessments and security audits.
- Protection of business information.
- Protection of sensitive information (by law)
  - Privacy law - for access to person registries
  - E.g. PCI-DSS - for access to credit card information
- SOA security
  - WS-Trust
  - WS-Security
  - WS-SecurityPolicy
- Participate in Disaster Recovery and continuity activities.
- Advocate security principles, best practices architectures, tools and processes.

#### Main activities are

- Designs more robust application security architecture.
- Assists in designing the application security architecture.
- Design security principles for service implementation and drive adoption of best practices.
- Designs the enterprise security infrastructure and architectural topology including recommending hardware, operating system, software, and information security requirements to ensure the confidentiality, integrity, availability, and privacy of information systems.
- Coordinates technical design/review activities with various groups including application development, enterprise architecture, information security, systems, network, and database groups to develop secure frameworks and enterprise applications. Provides access to the tools and technology needed to complete data integration development and overall data security.
- Collaborates with business management to communicate security risk and countermeasures.
- Researches, recommends and implements changes to procedures and systems to enhance security.

#### [Back to Contents](#)

- Creates security incident response plans, including coordination with appropriate departments, other Business Units, and appropriate authorities.
- Evaluates systems and procedures to safeguard internal information systems and databases from unauthorized users.
- Coordinates technical design/review activities with various groups including application development, enterprise architecture, information security, systems, network, and database groups to develop secure frameworks and enterprise applications.
- Collaborates with outside consultants/agencies as appropriate for independent security audits.
- Educates customers to increase awareness of information security policies and best practices.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

## 5 Delivery Models

Due to the nature of Integration projects i.e. in contrast with software development project, we have identified several possible agile delivery models. Those are Artifacts Driven Model, Kan-ban Model, and Sprint Model.

The Artefacts Driven Model is meant for smaller projects, usually lasting for 60 days maximum whereas the Sprint Model is for the project lasting more than 120 days.

We found that, in contrast to software development projects, integration projects requires a different approach in their *development* as well as *deployment*.

### 5.1 Artifact Driven Model

This approach is meant for smaller projects, usually 60 days max.

---

*We are presenting this model only because Integration Suppliers are forced to use it due to the lack of integration competence in organizations.*

---

Commonly organizations do not understand what artifacts are needed for integration. This leads to a situation whereby the organization asks artifacts from third parties one artifact at a time. In addition it's quite common that organization is not capable to evaluate the quality of artifacts. In Artifacts Driven model, the integration work can only start after ICC has received all reviewed artifacts required for design and implementation.

Too often Integration Supplier is assigned to do work without proper and complete artifacts. With proper processes, a mature ICC should not have the need to use this model.

This model is the most challenging for System Integrators and yet it is the most commonly used approach.

- The approach takes into account the fact that, in integration projects, very often artifacts are neither delivered in full nor correctly. This reflects the complexity inherent in most integration project.
- It also takes into account the possibility that, Customers might be in an early stage of their process development, therefore they might need an ample time to reach the maturity needed for integration projects.
- Mature ICC makes this approach futile.

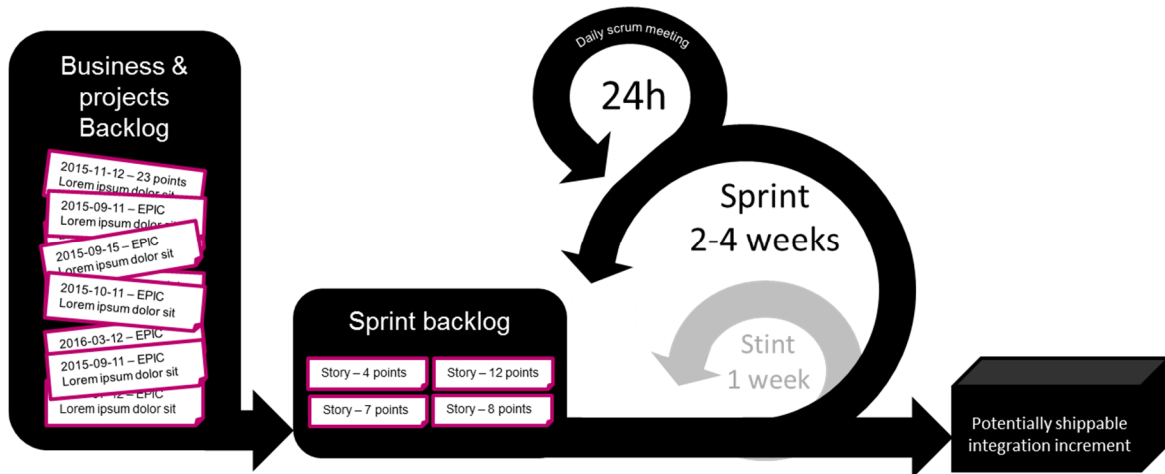
### 5.2 Scrum model

This approach is meant for bigger project, usually starting from 90+ days. Scrum model can be also run as continuous model where backlog is constantly filled. Scrum model is close to common scrum with few integration specific features.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)



**Figure 6: Scrum model with stints**

Scrum integration make progress in a series of sprints, which each deliver increment to existing integration solution. These increments may consist of integration solutions for whole new projects, changes to existing solutions and so on. Scrum model also introduces optional 1 week stint. As integration development tend to have more changes to requirements than traditional software development projects, stints are formal way of scheduling time windows for changes that were noticed too late. This coin has two sides: as good scrum master can now schedule unpredictable development and thus make realistic schedules based on e.g. earlier maturity of acceptance requirements it also teaches business owners bad habit – they might learn that they can iterate once with stints. Sometimes this is a necessity, but it is always a big extra cost. Putting stings to schedule takes also the rhythm away from the development team so use stints with extra care.

This approach has two main assumptions, which are;

- All artifacts required in a given Sprint are assumed to be ready at the start of the Sprint. If they are not ready and verified, the delivery model will degenerate to Artifact Driven model presented in previous chapter or missing artifacts generate unwanted stints.
- Business or Projects have to accept the fact that changes or new feature requests that are added during the current Sprint will be delivered in the next Sprint. This leads to a behavior where only critical fixes bypass normal *development* and *deployment* cycle. Quite common Sprint length in the field of integration is four weeks.

### 5.2.1 Scrum roles in ICC

Scrum is based on traditional software development and has a predefined role set for that purpose and specific set of artefacts like product backlog. Here is a general mapping of scrum roles and artefacts to ICC roles and artefacts.

scrum	ICC
<b>Product owner</b>	Product owner inside business unit which ordered a solution  In contrast to scrum, Product owner cannot dictate priorities of ICC as there may be several product owners from different businesses or projects. ICC Director and manager have that power.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

<b>Scrum master</b>	In smaller organization ICC Manager acts as scrum master, in larger enterprises typically separate scrum master is allocated for the project
<b>Sprint review</b>	Should be executed as in scrum
<b>Daily scrum</b>	Should be executed as in scrum (three questions: what did you do yesterday, what will you do today, is anything in your way?)
<b>Sprint retrospective</b>	Should be executed as in scrum
<b>Product backlog</b>	As in scrum, but in continuous development mode Product backlog is called business backlog and is continuously getting new stories to be implemented.
<b>Sprint backlog</b>	As in scrum
<b>Burndown charts</b>	As in scrum, only applicable for sprint backlog.

### 5.3 Kanban Model

This chapter illustrates how Kanban and especially Kanban tables can be used to continuously develop enterprise integration. It is intended for ICC which include the development function (Integration Supplier in earlier chapters). Following picture illustrates sample Kanban table. Development Kanban is similar to any software development Kanban, mostly the Definition of Done reflects the special concerns of integration development. ICC may and should adjust it with more phases if required.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

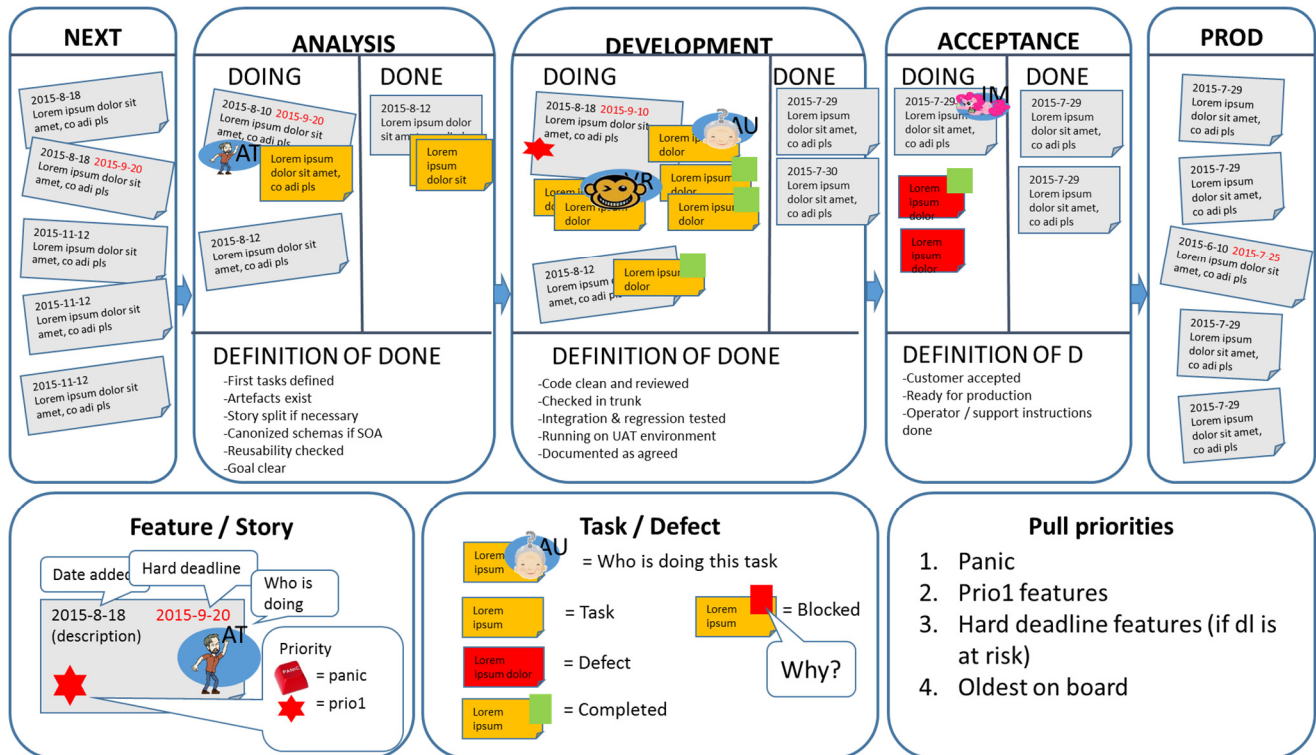


Figure 7 Sample Kanban table

Original software development sample: (Kniberg, Henrik 2009)

As backlog is evolving, column NEXT is filled with chosen stories or features to be developed. People attending development are pulling features for analysis phase where goal of the story or feature is cleared so that everyone on team understands it. Pulling occurs automatically based on predefined priorities shown in “Pull priorities” –list.

The story might be split if required and first doable tasks are defined. Tasks are also enriched with artefacts like mapping tables, business rules and so on. At this point it is also checked that whether we have this as a reusable service already. If we do, story or some of its tasks might go directly to Acceptance phase to ensure that existing SOA service really works with new story/feature as well. If service does not exist, the decision whether it should exist as a common reusable service (SOA) is done.

After analysis is done – and it is only done when it meets the Definition of Done – it is pulled to next phase with same priorities. Again a new set of Definition of Done requirements dictates when Development phase is ready and feature can be pulled to acceptance phase.

This approach is suitable for continuous development, projects and change management.

Things for ICC manager to consider with Kanban team

- The wider area of phases each participant can do, the more efficient Kanban team is as people can move to do those phases where there are most work to be done. For example, if developer can also work as analyst and functional designer, less bottlenecks tend to occur as work force is pulling them to phases where there are most features in queue.
- Ensure that team does not do anything outside Kanban table.
- Ensure that your Definition of done reflects all the best practices your ICC has chosen to do. For example using TDD or automated testing requires that automated tests are

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

ready after development. Code reviews require that review has left some kind of mark or documentation that it has really been done and so on.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

## 6 ICC Activities

To ensure that all tasks get done properly they must be closely controlled. Here we list which tasks should be executed more often and which can be dealt in a longer cycle. At first, we illustrate common activities related to *integration development*

The figure below shows the flow of activities with respect to ICC.

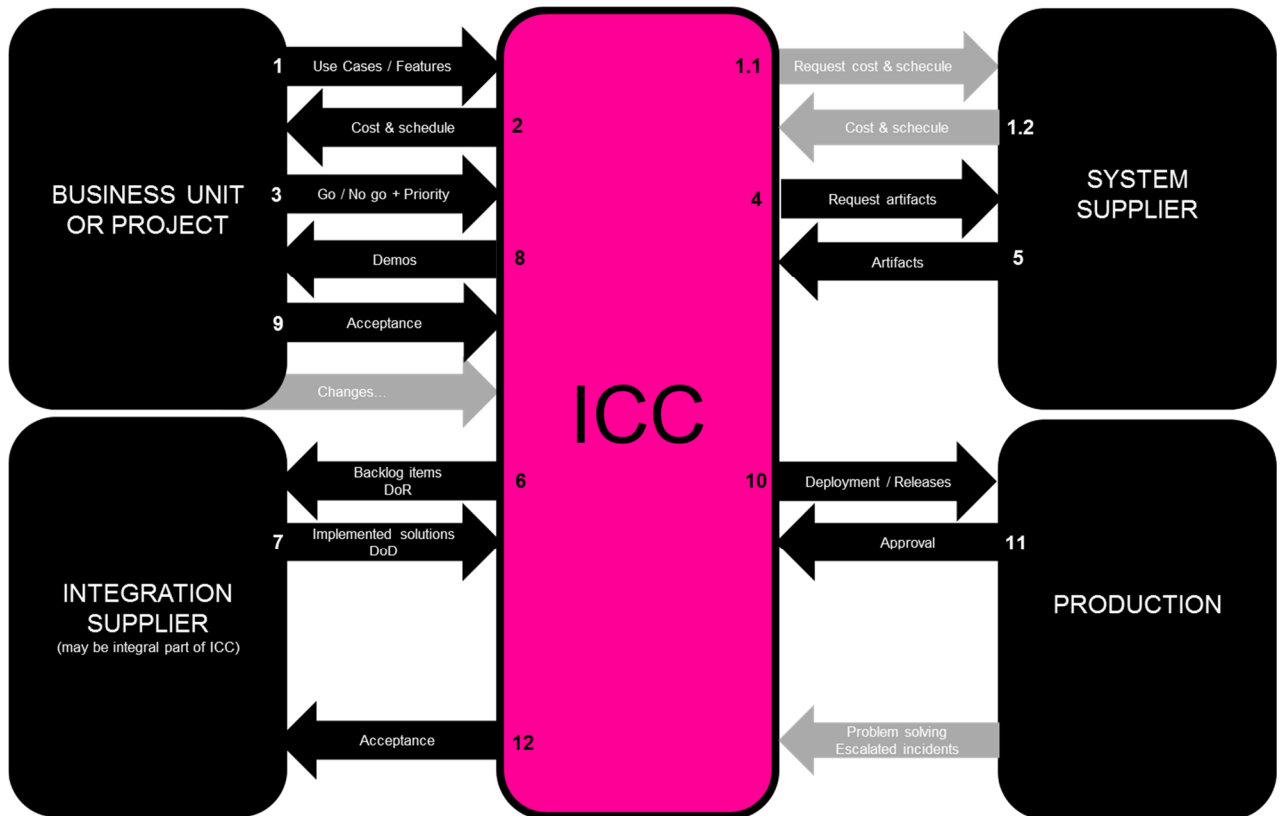


Figure 8: Activities Flow

### 6.1 Recurring Events

To ensure that all tasks get done properly they must be controlled tightly. Here we enlist which tasks should be executed on weekly basis, and which to be covered monthly. Mandatory roles that should be present in following meetings are also set here as an example. Roles that are required in meetings vary between different organizations.

#### 6.1.1 ICC Meetings

ICC meetings should occur at least twice a month - preferably, as often as the pace of development. Their target is to ensure that existing backlog items get done as well as new ones get in. ICC meeting also ensure that quality of backlog items is good or adaptable.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

Note that contents and pace of development meetings is heavily dependent on delivery model currently in use.

MANDATORY ATTENDANCE	OPTIONAL ATTENDANCE
<ul style="list-style-type: none"> <li>▪ <b>ICC MANAGER</b></li> <li>▪ <b>TEST MANAGER</b></li> <li>▪ <b>FUNCTIONAL DESIGNER(S)</b></li> <li>▪ <b>PROJECT OR SERVICE MANAGER FROM INTEGRATION SUPPLIER</b></li> </ul>	<ul style="list-style-type: none"> <li>▪ All other roles</li> </ul>

The meetings should cover and document following topics

1. Review of previous Development Meeting
2. What is the status of ongoing work; Status of the Technical and Business Backlog and their artifacts?
  - A. Solve problems if any
  - B. Make sure that artifacts are ready
3. Which are the new business requirements? If any, allocate Functional Designer and Architects accordingly. There may be a need for Business Owner to be present to demonstrate the requirement. Find a Process Owner for the new processes. Save the new business requirements into Backlog.
4. Define priorities for new Backlog items.
5. Allocate resources for new backlog items
6. Acceptance to Production (e.g. review to tests results)
7. Incidents from Production
  - A. Allocate resources for problem solving and update backlog items accordingly
  - B. Problem management - agreed changes to decrease amount of incidents
  - C. Lessons learned from transition (amount of incidents after PROD deployment etc.)
8. Communicate what is going to be deployed in next *deployment cycle* (see page 44). Are we forced to leave something out? Which INTS and deliverables are sufficiently tested? The contents and quality of *deployment package* MUST be approved by **ICC Manager** or **ICC Director**.
9. Changes and incidents from other organizational units or enterprise systems that may have an impact on Integration

Name	Description	Contact
Identify System	...	
Identify System	...	
Identify System	...	
Identify System	...	

10. Write a meeting memo and publish it to all stakeholders.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

[Back to Contents](#)

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

## 6.1.2 Steering Group Meeting

The Steering meeting should occur once a quarter, depending on the pace of development and number of possible Vendors.

The main purpose of the meeting is to synchronize all on-going activities in a particular project within [ICC Organization](#) with the aim of achieving the agreed milestones on ICC level. Also ICC capacity should be agreed here

Therefore, this meeting is responsible for the information exchange between groups.

The host of this meeting should be the ICC Director or the IC Manager.

Mandatory attendance:

- [ICC Director](#)
- [ICC Manager](#)
- [Test Manager](#)
- [Security Architect](#)
- [Integration Architect](#)
- Project or [Service Manager](#) from Integration Supplier

Optional attendance:

- [Technical Designer](#)
- [Business Information Specialist](#)

## 6.1.3 Lean Session

The “Lean Session” aka ‘retrospective’ is a meeting which main purpose is to eliminate waste in all other processes. By waste we mean steps in the processes that actually aren’t resulting any real value. Some of the waste is obvious, some forms of it hide them carefully behind words that sound important, and some forms of waste are just bad habits. In the other hand, some process steps are annoying and may not seem to provide any real value but are mandatory because they prevent waste from emerging later. For example creating automated unit tests during development takes a lot of time, but pays back afterwards. Read more about eliminating waste in Best Practices chapter.

In agile development process there is a session called ‘retrospective’ – ICC’s Lean Session does the same – but not just to development sprint like in agile software development, but the whole set of processes ICC itself has chosen to run. The Lean Session should be held on the first week after at least two “Deployments”

During the Lean Session identify and eliminate waste. Following checklist contains a list of questions that typically reveal the waste. Every time integration team finds waste, first ensure

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

it is really waste. This means that the specific process step will not provide value for development team or any other stakeholder during the life-cycle of integration solution at hand.

- 1) Do we have unclear requirements?
- 2) Are artefacts missing?
- 3) Are developers constantly switching between tasks during the day?
- 4) Did we start more than we can complete?
- 5) Are operators choking on the amount of futile alerts from our integration solution?
- 6) Do our developers have to wait for something?

If you find waste, ensure and eliminate it. Eliminating waste is actually a change in the process itself or in actions of other participants that for example gather artifacts and create requirements before development process.

### 6.1.4 Disaster Recovery Test

Disaster recovery test should be done at least twice a year.

Mandatory attendance:

- [ICC Manager](#)
- [Test Manager](#)
- Production [Service Manager](#)
- Project Manager from the Integration Supplier

Optional attendance:

- All other roles

The meeting should cover and document following topics

1. Every half year Test Manager MUST arrange Disaster Recovery Test according to Disaster Recovery Plan
2. Write a meeting memo

## 6.2 Communication Gap

Imagine a situation where a business representative is arranging a business to business messaging between own organization and the customer. For example, he or she must agree how the customer is sending their electronic orders directly from their ERP to (our) own ERP. Usually the technical part is managed by integration platforms at both sides.

Quite often when a requirement arrives to IT department from Business Unit or Project, the dead-line for delivery of solution is already set. The go-live date for business-to-business integration may be even promised to end-customer without even asking cost estimate or possible schedule or either of these from IT department. Imagine the chaos that these

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

promises given to end-customers lead when agreed with several end-customers without centralized control:

- Solutions are deployed to production based on uncontrolled schedule; Cyclic-deployment is impossible and therefore existing integration solutions are vulnerable to unexpected side-effects from hastily deployed new solutions.
- Due to the dead-lines not based on reality, solutions are implemented and delivered in haste. It is quite common that proper testing and test-cases are compromised in the name of hurry.
- Resourcing for information gathering in own organization and implementation in Integration Supplier cannot be arranged in either of these parties in time. This leads to end-customer promises that cannot be kept and that leads to decrease in end-customer satisfactory.

ICC must end this careless activity. This is achieved by formalizing the way communication is handled between end-customer, business, ICC and with those Integration Suppliers that ICC controls. Key rules in formalization are

- Business people ask Cost estimation and schedule from ICC before making a promise to end-customer.
- ICC ensures delivery dates and costs from Integration Suppliers and people responsible of maintaining the test and production environments.
- ICC provides documentation or fact sheet what kind of a business to business integration is possible and what is not. This document describes what integration services already exist for fast and cost-efficient integration.
- Centralize communication with suppliers and third parties to ICC

When setting up these rules, remember that ICC's and whole IT organization's purpose is to serve Business needs better, not to make Business agreements harder by bureaucracy in internal communication.

Therefore communication between ICC and Business people must be kept as easy and fast as possible, but still preserving the key rules: what is promised and on what basis.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

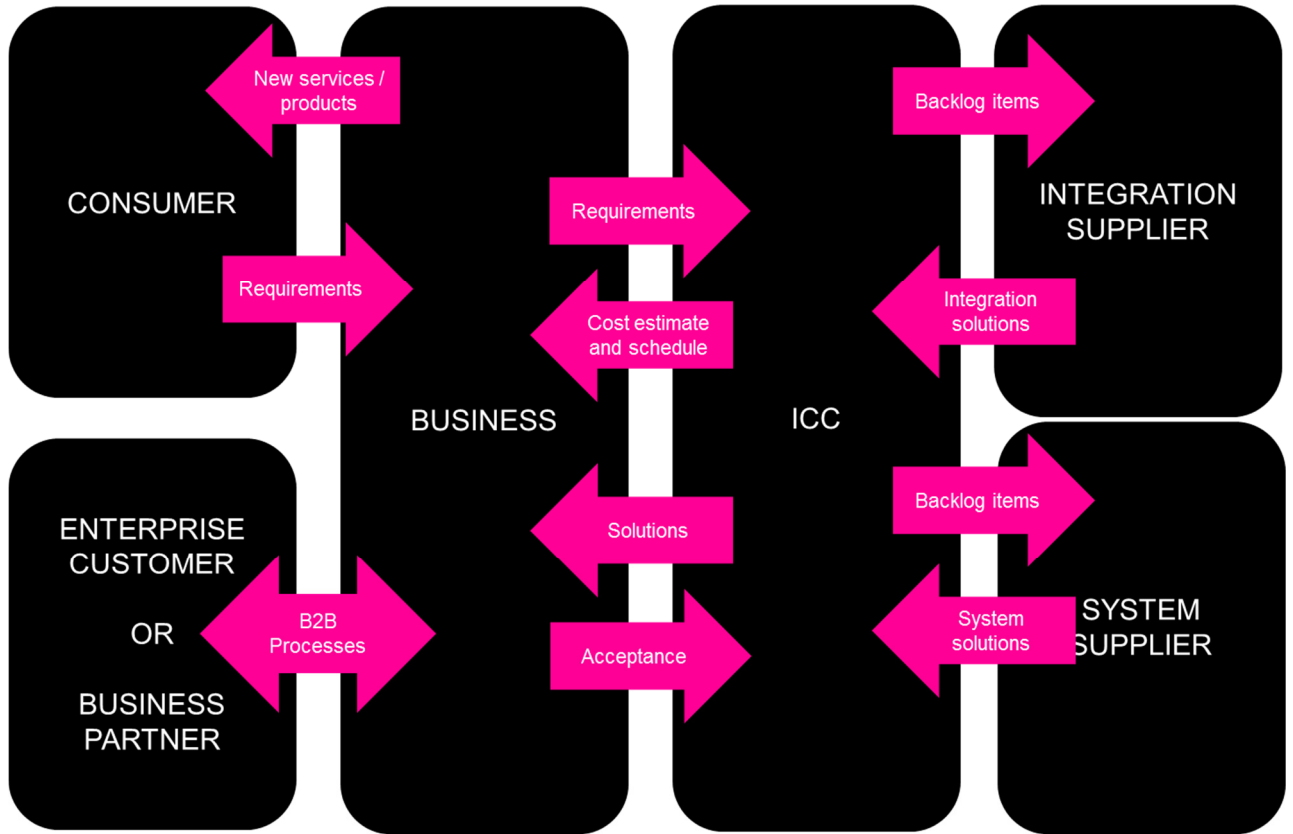


Figure 9: Communication flow

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

## 7 Best Practices

Our aim has been to develop and therefore contribute to a framework that guides and encourages the development of best practice in integration projects.

The whole process of developing a best practice is partly about making sure that right questions get asked and answered and partly about making sure that all parties understand these answers.

And last but not least, is the part which consists of helping people to develop the right answers.

This chapter outlines certain project's aspects which are common in our integration work.

### 7.1 Eliminate Waste

By waste we mean steps in the processes that actually aren't resulting any real value. Some of the waste is obvious, some forms of it hide them carefully behind words that sound important, and some forms of waste are just bad habits. In the other hand, some process steps are annoying and may not seem to provide any real value but are mandatory because they prevent waste from emerging later. For example creating automated unit tests during development takes a lot of time, but pays back afterwards. Also quite often waste is a result of neglect in previous process steps or previous processes.

As Toyota Production Systems created Lean manufacturing originally, they identified 3 forms of waste:

- 1) Muda – unproductive
- 2) Mura – unevenness, inconsistency
- 3) Muri – over-burden, unreasonableness

Tom and Mary Poppendieck derived these manufacturing wastes into lean software development. Their samples of waste contained things like unnecessary code or functionality, starting more than can be completed, partially done work, task switching, quality issues, bugs and so on.

As we go in to more and more specific area, we can identify wastes more accurately. Following list is a common list of wastes in process integration that arise during development process.

- A. **Missing artefacts** – this occurs during development phase when developer notices that he or she does not have complete mapping chart, test cases or other mandatory items we call artefacts.
  - Solution: Definition of Ready (see chapter Definition of Ready and Definition of Done)
- B. **Changing or incomplete requirements** – this occurs during development phase when developer already has started to work, but is halted due to a fact that requirement that initiated backlog item, wasn't complete.
  - Solution: Requirement management
- C. **Constant task switching** – if developer is required to change tasks constantly during the day the context switch eats efficiency and morale which again

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

reduces performance over time. Task switching may occur due lots of reasons, most common reasons are

- work is divided into too small chunks -> solution: tune your backlog and work habits
  - production incidents escalate to developers -> solution: go for real DevOps, train operators and give them accurate and inclusive instructions, automate recovery processes in production
- D. **Redundant work** – this occurs usually in SOA based service development where underlying system – the target of integration – is also refactored or changed during integration work. Quite often the system supplier tends to to data-mapping etc. work directly under the hood resulting in redundant work or functionality in wrong level of the architecture.

Concepts like Requirement Management and Definition of Ready can be used to prevent the “rework cycle” by getting the requirements and other artifacts as early as possible.

ICC should execute constant “waste elimination”. This can occur during sprint retrospective or monthly meetings depending on the delivery used in analysis and development. It is crucial, that ICC manager or ICC director also participate to these session and remove waste generated before development. For example, changing requirements is usually fixed at the point where requirement are created, which quite often occurs before development. This said, analysis phase of Kanban model presented earlier can also fix this problem during Kanban development.

## 7.2 Service Oriented Architecture

SOA is one of the fundamental architecture patterns of system and process integration. As the name says, it is based on the concept of a Service. In integration development, it can be thought as a design pattern where aim is to implement business activities as reusable services.

A Service

- is an implementation of logical business activity (getCustomer, checkCredit, createOrder etc.)
- may be composed of other services (composite-component model)
- is self-contained
- is loosely coupled
- is reusable
- is a black box to consumers of the service

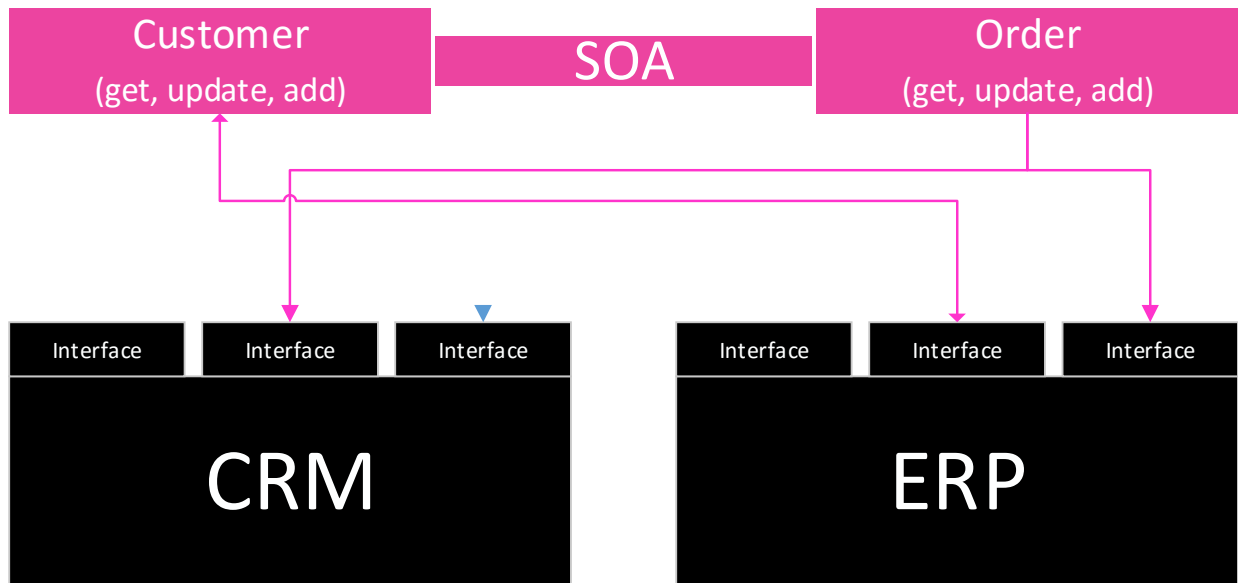
It is quite common misunderstanding to confuse interface to a real SOA –level service. For example, a line-of-business system vendor may state that their CRM has SOA –level service *getCustomer* as it returns customer information. This is just an interface as it does not necessarily contain all the information that company has identified elements if concept “Customer”. It surely does not have the format and structure that company has canonized for itself (see chapter Canonical Enterprise Data, page 55).

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

SOA itself doesn't specify any standards or technology. As integration platform is a centralized system and connected to other systems and trading partners, integration platform vendors often include service publishing capabilities as well making them ideal SOA publishing platforms.



Gartner have studied that service reusability brings 30% cost savings. As Integration Competency Center gathers the requirements from business and their choreographed processes, ICC must always identify business activities that are reusable and can be self-contained and are reusable. As services are implemented, their governance is also a fundamental work for ICC. SOA governance include following topics for ICC to handle

- Service catalog
- Quality of services
- Service policies to restrict behavior
- Performance monitoring
- Managing how and by whom services are used
- Compliance to standards
- Use of canonized data
- Managing service lifecycle

In addition to SOA governance, ICC takes care of following tasks

- Identifying reusable business services (Business Analyst or ICC architect)
- Ensuring reusability (ICC architect)
- Ensuring SOA requirements listed in the beginning of this chapter (ICC architect)
- Keep service granularity inline: avoid single do-it-all mammoth services, avoid tiny function-like services
- Ensure that a service is a business activity, not an interface for individual line-of-business system

[Back to Contents](#)

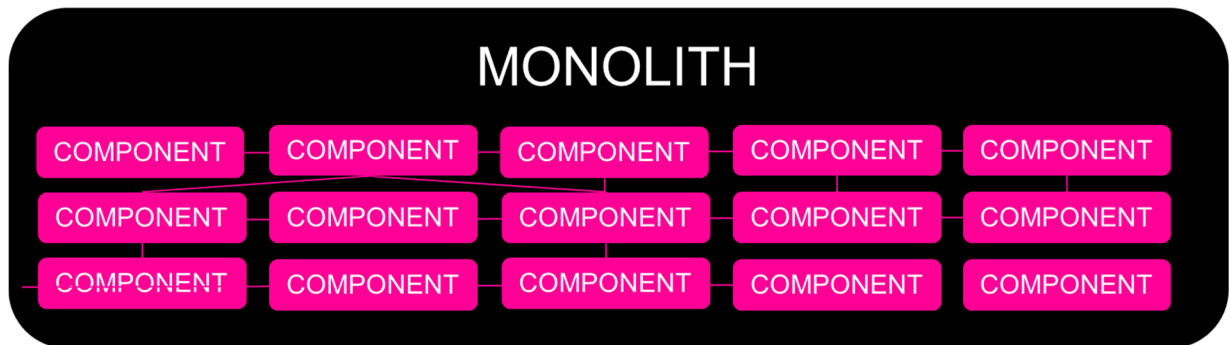
ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

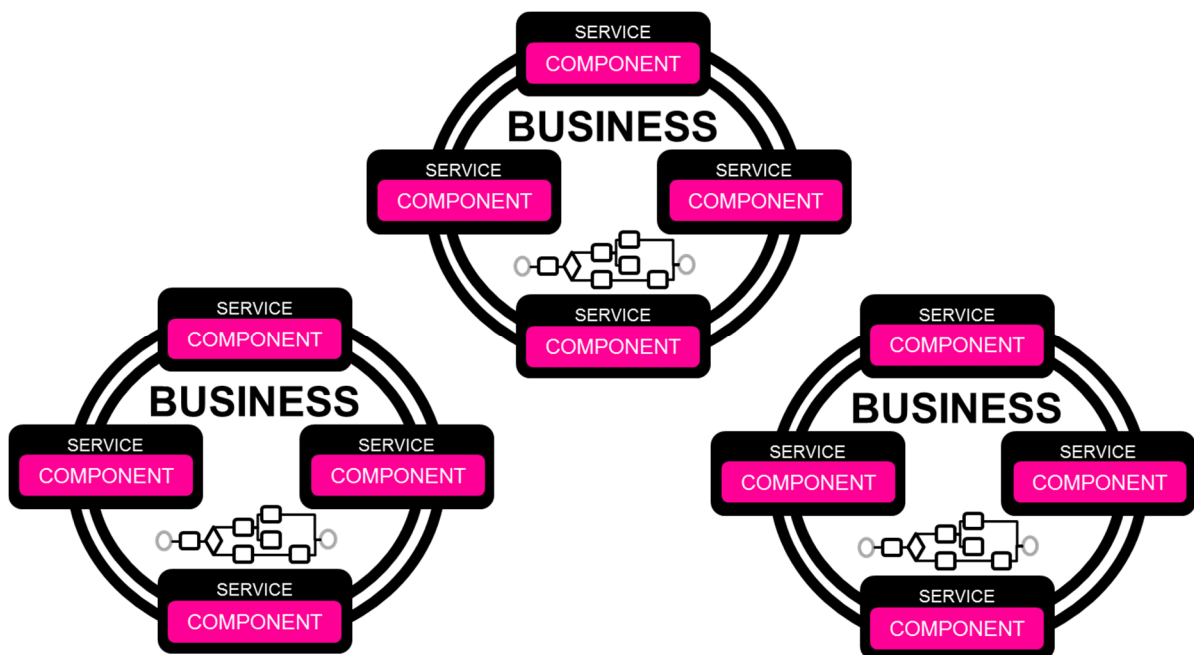
### 7.3 SOA and Microservices

Quite common topic of discussion is whether the new [Microservice Architecture](#) is the successor or even a pattern that makes old SOA obsolete.

To understand Microservices architecture, first think of some huge monolith application with wide variety of functionalities as in picture below. Typically, these components have references to each other and they cannot be considered independent.



Split the monolith to a set of components that are independent in functional and technical sense. This means these components are independently replaceable and upgradeable. Publish each component as a service (e.g. REST) and let them evolve around business capabilities (e.g. business units) as in picture below.



This setup still follows patterns of SOA, but the service orientation, granularity and different development cycles (of for example business units) are also taken into account. Microservices Architecture also emphasis on strong decentralization – this allows fast on-demand service development yet requires even more governance. In this context, ICC’s SOA governance can act more like a librarian and marketing function for already existing services – reusability of existing services should be enabled and recommended yet not forced.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

Microservices architecture should be seen mainly as a software architecture yet it also solves great amount of classical SOA caveats.

## 7.4 Cyclic Deployment

Cyclic Deployment is an activity whereby deployment is done continuously.

The word "Cyclic" refers to prescheduled deployments that may occur for example every four weeks.

This chapter explains the meaning of Cyclic Deployment. A set of new functionalities, changes to existing solution and bug fixes are managed as *deployment package*. Later in this document we shall call these new functionalities, changes and fixes just "modifications". Modifications that are depending on each other must be put into the same Deployment Package.

Several functionalities may build up to be an implementation of a Use Case: therefore a *deployment package* can contain one or more Use Cases.

A *deployment package* is not related to any specific project – it may contain modifications for several projects, these modifications may be in the form of important fixes as well as changes to previously deployed functionalities. The *deployment package* is not related to any delivery model, therefore it can be deployed at any time whenever deployment is possible. The key point is that the *deployment package* should be thoroughly tested.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

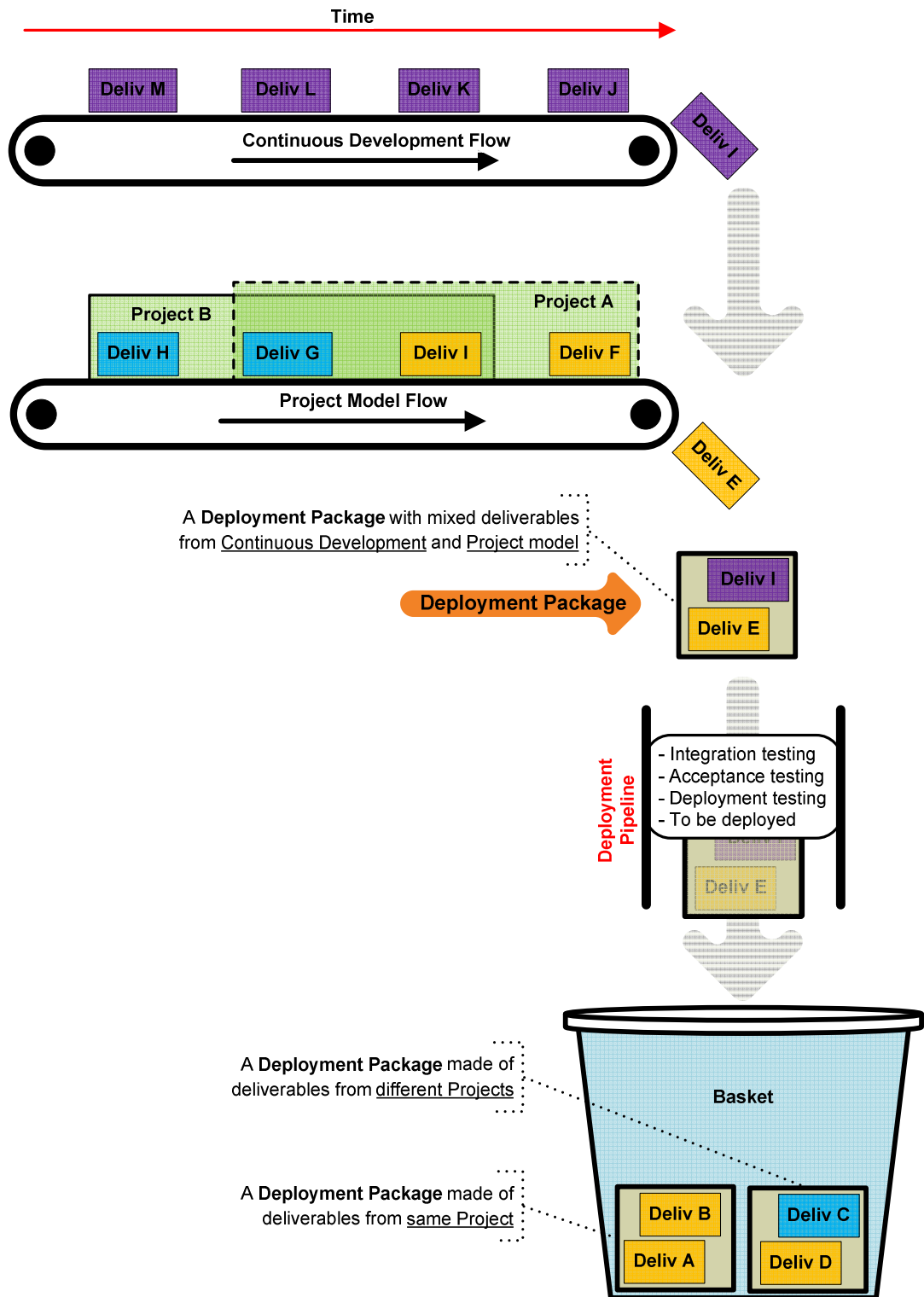


Figure 10: Continuous Development process

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

As was said in the beginning of this chapter, **deployment package** is deployed as a whole. It is also going to be integration tested as well as accepted by internal or external customer as a whole.

Integration tests MUST be accepted before proceeding to **deployment process**. Acceptance is done by **ICC organization** as part of the **deployment process**. In the other words, **ICC** is

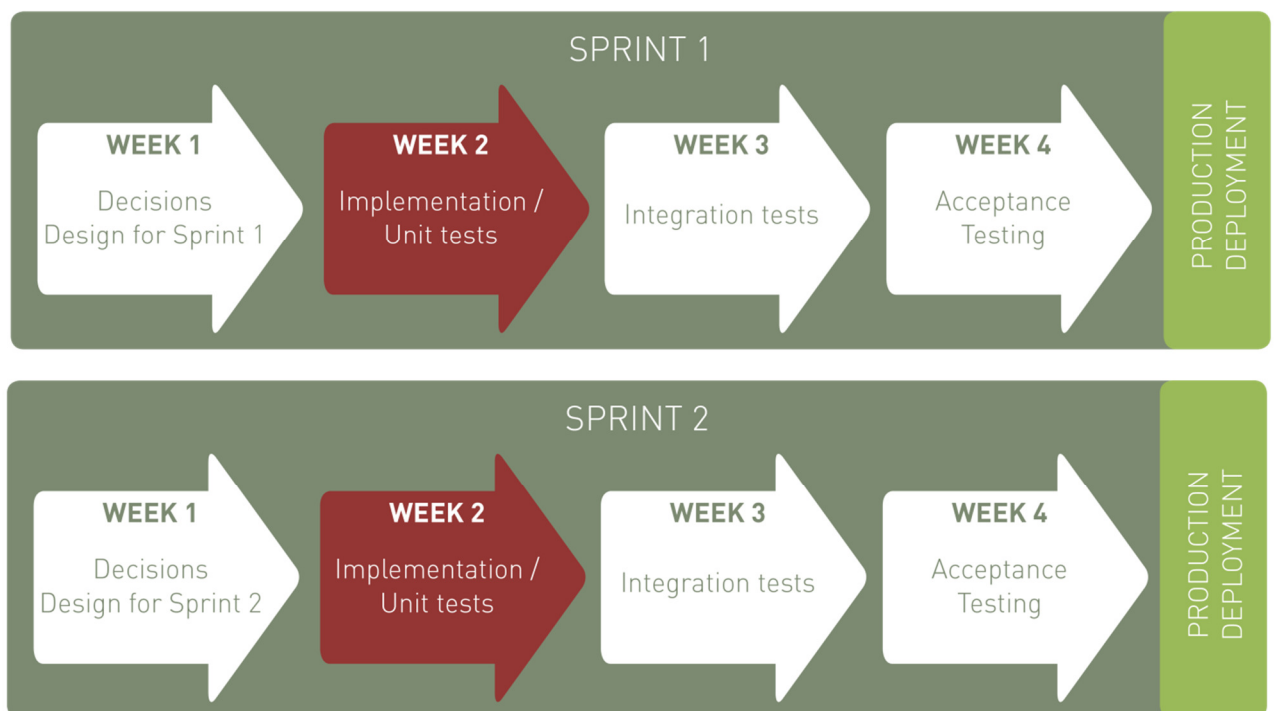
- Accepting the **deployment package** as a set of functional solutions
- Accepting the quality of the contents of **deployment package**

Not only the test results must be acceptable but also the Test Cases MUST be approved by ICC's Test Manager; each Use Case MUST have more than one successful Test Cases and at least one test FOR EACH error condition.

Every **deployment** MUST produce documentation, based on formalized template, e.g. Agile Integration Model (AIM).

One common error is to combine **deployment cycle** with **development cycle**. By sequencing **deployment** and **development** steps inside a single time box of four weeks, the time left for implementation is reduced to one week.

Following picture illustrates sequencing problem.



**Figure 11: Sequencing problem**

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

To ensure that Integration Supplier(s) work efficiently, ICC must separate the ***deployment process*** from the ***development process***.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

To solve the sequencing problem, **deployment** and **development** processes need to be separated.

Both processes should be run concurrently. To achieve this, the Sprint (**development**) and **deployment** are run in two separate cycles. These cycles are aligned together in between Sprints.

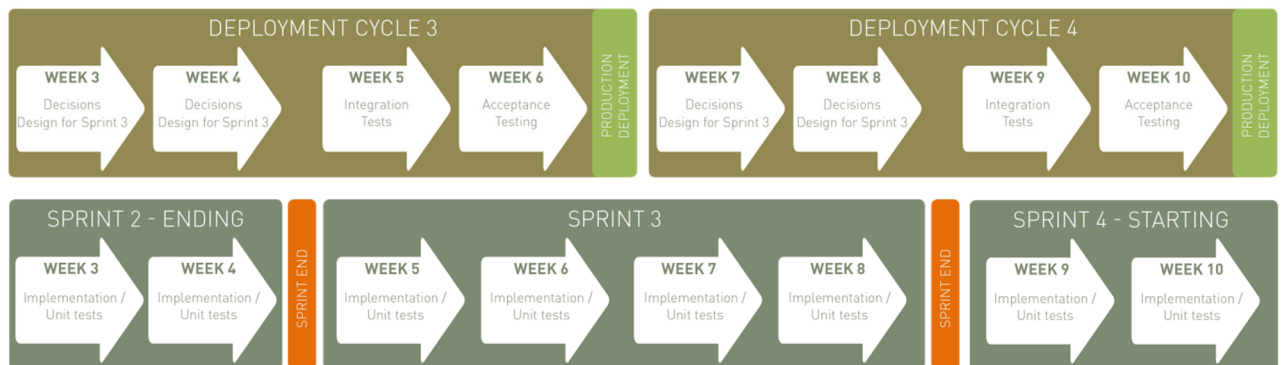


Figure 12: Development and Deployment Cycle

To ensure that there is enough time for Integration tests as well as Acceptance tests, the contents of **deployment package** in this example are frozen two weeks before the **deployment** date.

The contents of next Sprint should be decided before the current Sprint is ended.

## 7.5 Testing

It is easy to agree that testing is a best practice – a mandatory part of multiple phases of **development** (unit testing, system testing) and **deployment** (integration testing, acceptance testing, deployment testing). Following list covers the most common measures to avoid pitfalls in testing of an integration solution.

- Understand that testing is not something that happens once during *development*. Testing is done at least during *development*, *integration* and *deployment*.
- Acceptance test cases must exist before development starts
- Test cases MUST be derived from Use Cases / User stories
- Unit tests must be automated
- Aim to automate Acceptance tests as well, this leads to better quality
- Apply test-driven development and Acceptance test-driven development when possible
- Error conditions of Test cases MUST be defined within the Use Case.
- Technical Designer may add test cases which he or she thinks are mandatory but which are not actually testing the functionality itself but the way the functionality was implemented.
- Test material must correspond to real-world scenarios.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

- Stress tests must be done if any functional or non-functional use case indicates the need for Stress testing.
  - Bypassing Stress testing should be based on facts and NOT assumptions.

## 7.6 Test-driven development

**Test-driven Development** (TDD) is one of the best ways to ensure quality of unit testing in integration development. The idea is to first create automated test cases. The first test round should fail. This failure occurs because the developer has not yet implemented the integration solution that leads to success and ensures that test cases work correctly.

Developer can start his or her work after the all paths of process execution have been covered with automated test cases. Each test case should turn green to illustrate that the integration solution under construction is coming closer ready status.

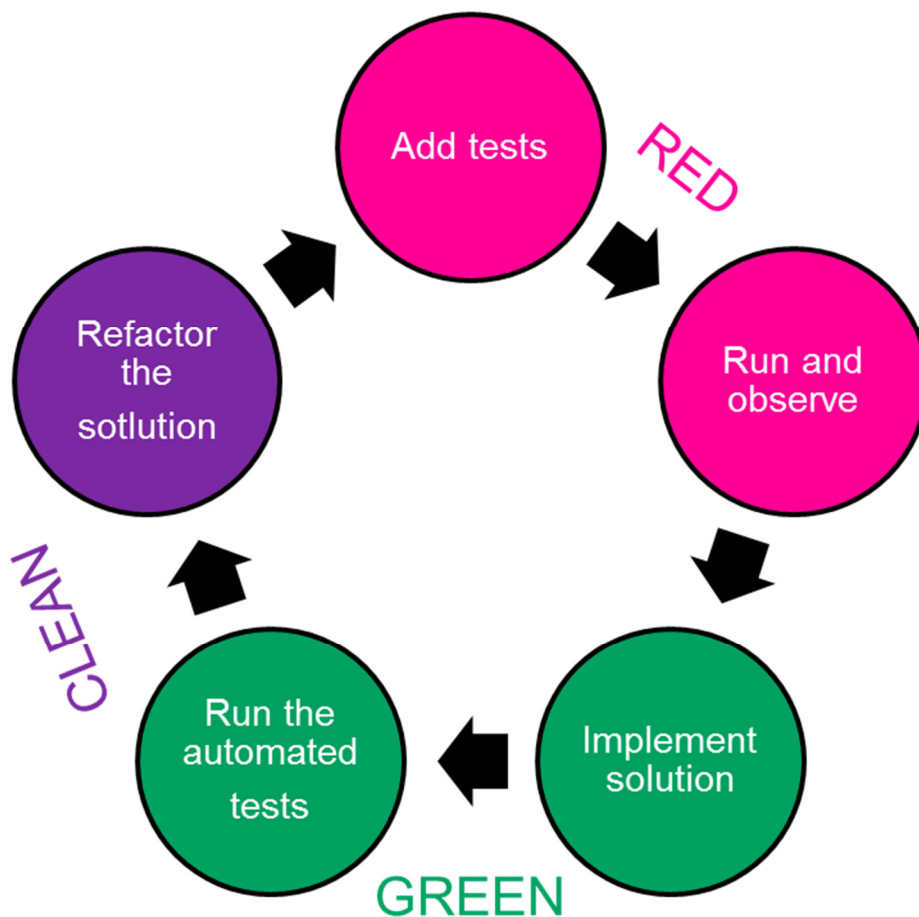


Figure 13: Test Driven Development Cycle

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

When considering **TDD**, pay attention to following issues

- TDD requires test automation software compatible with the **Integration Platform**. Don't make the need for automation platform as a excuse; test scripts or code can be run even on the integration platform itself, if separate automation system is not available.
- TDD is not possible with all **Integration Platforms**. Therefore, Find out what kind of tools exists.
- Defining and creating automated test cases adds costs, but radically enhance quality. Because of the costs related issues, management support is essential. Meanwhile, management should consider TDD as an investment to quality which eventually reduces total costs.
- If the integration solution, which is to be tested automatically, includes external communication, e.g. **Web Service** call; those external interfaces must be present as mock-ups or test interfaces of other systems. Mock-ups must write their own simple messaging trace logs: what came in from integration process and what was mock-ups response.
- If the integration solution under development shall change a persistent state in an external system, then the external system should be automatically reset before retesting. For example, an external database's data is initialized to have certain values before automated test is rerun.
- Beware of process boundaries - originally TDD was developed for unit testing. When external systems are included in test cases, the testing easily overlaps integration's integration test. Therefore a failing test case may occur due to a failing external test environment. A situation which not common in traditional software TDD.

### 7.6.1 Acceptance Test-Driven Development

In ATDD whole team - including the party who originated the requirement - collaboratively defines acceptance criterias and then create a set of concrete acceptance test cases. It's the best way to ensure that product owner (business) and the developers have shared understanding what the integration solution must actually do.

Just like in TDD, test cases are automated before development and as developers implement the solution. Solution under development constantly achieves partial acceptance as each automated case is passed until it is ready - and fully accepted by criterias and cases all parties undersigned in the beginning. TDD and ATDD cycle is illustrated in figure below

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

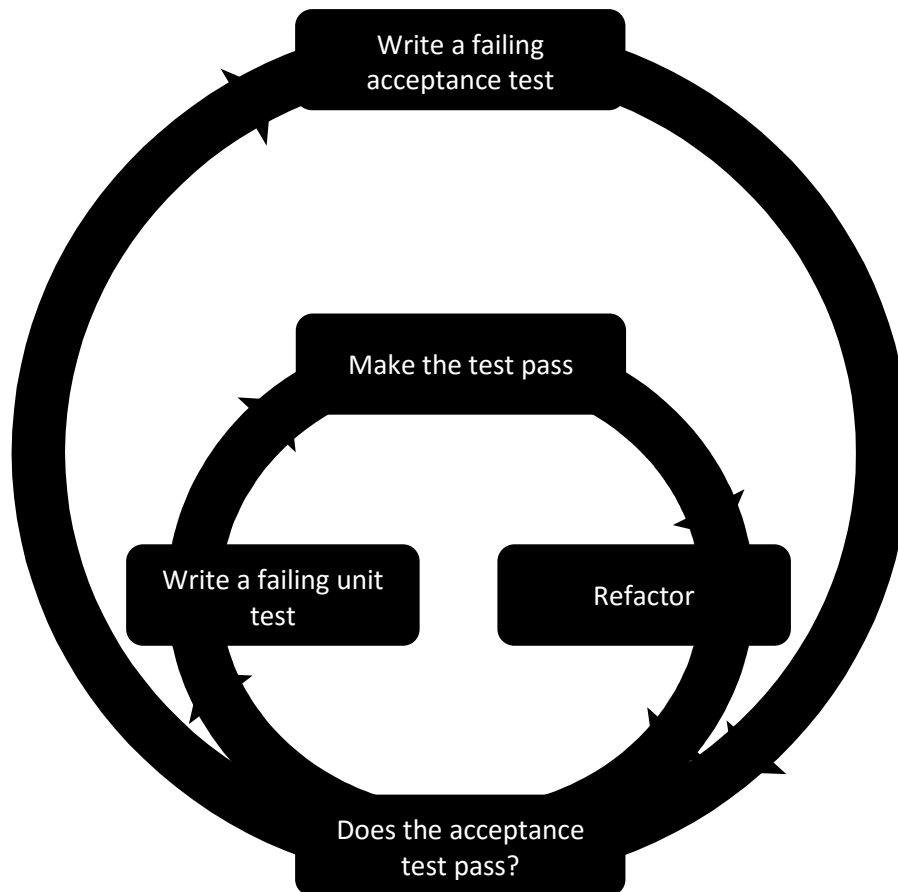


Figure 14 ATTD with TDD

## 7.7 Artifacts Review

All internal and third party artifacts should be reviewed by [ICC](#) before sending them to Integration Supplier.

The review should include following steps and tests:

1. Ensure that **functional** and **non-functional requirements** exist and are comprehensible. It is not enough to tell what it should do – developers need to know how fast for example service should answer, how much it is stressed simultaneously and so on.
2. Ensure that **process exceptions** are listed and defined what to do in each of them. From 50% to 80% of integration developing time goes to exception case handling. This does not mean similar exceptions as in software but process level exceptions.
3. Ensure that **source** and **destination data schemas** exist and are comprehensible.
4. Use a versionable mapping table.
  - Mappings tend to change more often than the process logic; make sure that mapping table (e.g. excel document) has element level version control.
5. **Identify** 3rd party **interfaces** that are risky. Following reasons increase interface's risk level
  - Interface is created for this need. In other words, it is used for the first time.
  - 3rd party documentation, schema or other artifact is missing.

[Back to Contents](#)

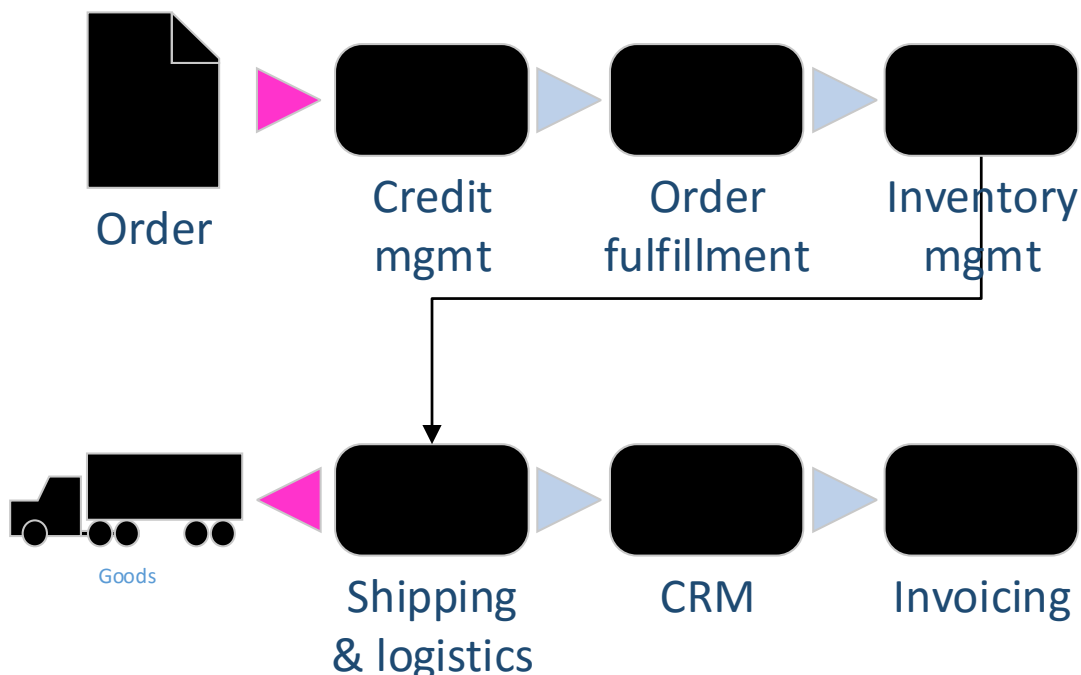
ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

- Documents prove that interface is not following other de-facto standards properly. Common example of this is an interface that is reading in XML -structures but requiring for example line feeds. It means that XML is read as a flat file and record separator is something else than an end-element.
6. Require full documentation of external interfaces.
    - Integration Supplier needs all the information of interface that is used to pull out or push in information.
  7. If XML is used, require XML-schemas.
    - Common reason is that they may read XML in, but due to the age of the legacy system, they do not read it by using proper XML-processor.
    - If a 3rd party cannot deliver an XML-schema there is typically something wrong.

## 7.8 Canonical Enterprise Data

Canonical enterprise data is a goal where all enterprise data is structured, formatted and delivered in unified format and semantics through the enterprise. Enterprise data means all the concepts used during executing the business processes are identified, named and structured in unified manner. For example, company has an order to cash (O2C) high-level process where company receive orders from customers, fulfill them and invoice them contains reusable concepts like Customer, Order and Invoice and so on. If company canonize concept Customer, it means that company agrees to use exactly same attributes in exactly same meaning with Customer data across processes.



**Figure 15: Sample O2C process**

As you can see, there are several concepts that should be canonized. Let's focus on the concept Customer. First, when the order is received by e.g. "receiveOrder" SOA service, we initiate the O2C process. After initialization we query SOA layer's service "getCustomer" which returns us

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

the canonized Customer information. The first step is to identify Customer and ensure check credit status of customer. As we have always identical schema for Customer, process already know how to read VAT number from Customer element that “getCustomer” service returned.

```
<Customer>
  <VATnumber>FI07654321</VATnumber>
  ...lots of information...
  <DeliveryAddress>
    <Street>Vaisalantie 6</Street>
    <PostalCode>02160</PostalCode>
    <City>Espoo</City>
    <Country>Finland</Country>
  </DeliveryAddress>
  ...lots of information
</Customer>
```

As the process goes on we need delivery address, billing information, updates to CRM and so on. All required information can be found from Customer element. The point of canonization is that any other process that need Customer data, can always retrieve it in format agreed beforehand.

Canonization occurs also on several levels of data. For example, company may canonize subdata elements like <insurance> and what data and attributes contain and then use it inside other high level data elements like <employee> or <customer>.

Benefits of canonization are obvious: reusable services contain reusable data elements, data mapping is faster during integration work and so on.

Common question is that what to canonize? Everything? There are two different approach for this: huge data modeling project vs. process oriented modeling. In data modeling projects usually third party analyze enterprise processes and data and deliver inclusive documentation and schemas about all the relevant data elements through the company. Typically these projects tend to be long and tedious as there are enormous amount on concepts in larger companies. Process oriented modeling is run by ICC and it models those concepts it identifies to be reusable. ICC is already responsible of implementation of SOA layer, ICC must understand the process that integration platform is executing so it is quite easy to assign the responsibility of data modeling to ICC as well. It depends on the ICC how well it understands the business and concepts, so never forget the relevance of Business Analyst role.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

## 8 Conclusion

As you have seen, we have outlined our experience in establishing [ICC Group](#). In the first and second chapters we defined [ICC](#) and gave reasons as to why establish ICC as well as outline its benefits.

In the third chapter we went through the basic steps to establishing an [ICC Organization](#), we outlined the requirements as well as the challenges.

Chapter four lays out the [ICC Ecosystem](#) with its building blocks which is made-up of different roles. We therefore went on by describing each role and its responsibilities by applying our experience in numerous integration projects implementations. Our experience throughout years has led to our attitude of more strict definitions of roles and responsibilities in an ICC Organization. When developing a management model such as ICC, it's important that each role is precisely defined, such as, who is responsible and for what tasks. As we have seen, the following were the roles defined - ICC Director, ICC Manager, Information Specialist, Integration Architect, Functional Designer, Technical Designer, Project Manager/Service Manager, Test Manager and Security Architect.

Chapter five introduces HIQ-ICC [delivery models](#). Those are Artifacts Driven Model, Sprint Model and Kanban model.

In chapter six we introduced [Activities Flow](#). These activities act as a medium of communication among ICC's key members. The activities consisted of - Weekly Meetings, Monthly Meetings and Half-year Actions. Whereas the afore mentioned activities are more concern into day to day and the general progress of [ICC work](#), the [Communication Gap](#) is concerned with managing business communication between ICC Group and its Customers. Business communication must be controlled very closely by the ICC Group. Any failure in communication will lead to mismatch between Customer's expectation and ICC's commitment and/or vice versa. As we all know, this mismatch is a recipe for conflict.

Chapter seven introduces the [Best Practices](#) through several of the central activities in HIQ-ICC development model. Those are Cyclic Deployment, Testing, Test-Driven Development, Artifact Review and SOA. Deployment process requires a good deal of control, to accommodate this observation HIQ-ICC introduced the Cyclic Deployment process in its *development model* i.e. AIM - Agile Integration Model. Whereas Artefacts Review reflects the nature of integration complexity with respect to specific industry. Testing and SOA are best practices which any integration project big or small should never ignore. As we know, a small integration project will most likely grow big and even if it won't it will certainly need to be continually maintained; and if you want a smooth maintenance, then testing and cyclic deployment are your best friends.

To summarize, the [ICC Group](#) is responsible with different aspects of the whole integration effort as outlined here: *Integration Methodology* - process of running an ICC, defining it, organizing it, all the things you need to run an integration group, and how it will interact with other IT groups; *Integration Systems* - all the discipline of how you manage, plan and operate the system; *Modelling Management* - includes techniques around canonical data modelling, what are the best practices and how do you build them; *Architecture* - the information architecture; *Metadata Management* - ICC Group is responsible for data assets; *Business*

[Back to Contents](#)

ICC Handbook, HIQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

*Process Management* - this includes service flow modelling, information flows, business event modelling, and common definition of business events; *Financial Management* - ICC operates as a shared service and a shared infrastructure.

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

## 9 APPENDIX - A: ICC checklist

COMPANY LEADRESHIP/MANAGEMENT	
Is <a href="#">ICC</a> initiatives mapped onto Company’s objectives?	
Have responsible management approved the <a href="#">ICC establishment</a> ?	

ICC - GROUP	
Determine if the <a href="#">ICC Group</a> is established, i.e. <a href="#">ICC Manger</a> , <a href="#">Information Specialist</a> , <a href="#">Service Manager</a> and <a href="#">Integration Architect</a> .	
Are all members of ICC Group aware of ICC commitment?	
Do all ICC members understand the processes concerning <a href="#">ICC activities</a> ?	
Do ICC members understand the intentions and expectations of ICC Group?	
Does everyone involved in ICC Group understand their level of involvement, <a href="#">roles</a> , and responsibilities?	
Are project retrospective i.e. <i>post implementation &amp; reviews</i> used to learn from projects?	
Does the Company or Department have evidence of demonstrable and measured improvements towards excellence with respect to ICC?	

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

<b>PROCESSES &amp; SYSTEMS</b>	
How well defined are the processes below? How mature are those processes? - Development process - Deployment process - Incident Management - Change Management - Problem Solving Is there a need to introduce any of these processes?	
Are there formal mechanisms for developing and retaining ICC Group's knowledge, skills and experience?	
Does <a href="#">ICC Group</a> have mechanisms in place for recognizing and sharing knowledge from projects?	
Review processes to ensure that, the ICC Group is functioning as intended	
Are feedback and review, are processes integrated into Company's processes?	
What is the authorization process for moving an integration solution into the production environment?	
What prevents unauthorized changes to be made after the completion of testing, but prior to transfer to production?	
How does management provide oversight over the system development and maintenance process?	
Do developers have update access to the production environments?	
Is the testing environment sufficient to identify issues with integrated testing efforts?	
Is a separate test environment used for testing program changes?	
What is the change request process (for both in-house changes and changes performed by the vendor)?	
Who approves change requests?	
What documentation exists to support a change request?	
How are change requests tracked?	
Is any history maintained of all changes to the system?	
Is there a version control mechanism in place?	

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

What ensures that the source code changed is the most recent version of the program?	
How are Business Owner involved in <a href="#">ICC activities</a> ?	
How are emergency changes to production performed?	
Is a log of all emergency changes maintained?	
Is code review done?	
Is the code put into Version Control?	
How do business users authorize emergency fixes to production data or programs?	
Are there monitoring mechanism in place to ensure adherence to policies and processes?	
How the follow-up is done?	

**COMMUNICATION**

<b>COMMUNICATION</b>	
Is there a systematic <a href="#">communication</a> <i>within</i> and <i>around</i> <a href="#">ICC Group</a> ?	
Is communication between <a href="#">ICC Mangers</a> , Staff and Customer established and maintained?	
Is there active engagement with all Stakeholders/Partners <i>within</i> and <i>outside</i> <a href="#">ICC Group</a> ?	
How changes are communicated and approved?	
What are escalation procedures for major issues?	
How are policies and procedures communicated?	
What other tools are used for communicating and monitoring Integration Projects status and issues?	
How often does the team meet?	

**FOCUSING**

<b>FOCUSING</b>	
Examine the open issues list to ensure all significant issues have been addressed	

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

Lessons learned from project reviews are shared and adopted within the Company?	

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

## 10APPENDIX - A: Persons assigned to ICC roles

Unit	Role	Last Name	First Name	email	phone	Company
ICC	ICC Director					
ICC	ICC Manager					
ICC	Information Specialist					
ICC	Integration Architect (active)					
ICC	Integration Architect (on demand)					
ICC	Information Architect					
ICC	Functional Designer					
ICC	Security Architect					
ICC	Test Manager					
ICC	Service Manager					
ICC	Technical Designer					
Production	Service Manager					

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

# 11 References

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)

## 12 Bibliography

- Kniberg, Henrik. 2009. "Tools." *Blog*. <http://blog.crisp.se/tag/tools>.
- Wikipedia. 2008. "Scrum (development)." *Wikipedia*. Accessed 2010. [http://en.wikipedia.org/wiki/Scrum\\_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development)).

[Back to Contents](#)

ICC Handbook, HiQ-ICC 2015 → [www.unthink.fi](http://www.unthink.fi)

This work is licensed under a [Creative Commons Attribution-Non-commercial-Share Alike 3.0 Unported License](#)