



# Integration Competency Center

---

## Management Guide

*Version 2.1*

9 June 2010

Authors: Antti Toivanen, Gibson Jotham



ICC - Integration Competency Center

ICC is a shared service intended for companies who wish to design, develop and maintain integration solutions that is serving more than one business units or projects

---

**FIGURES..... 2**

**INTRODUCTION ..... 3**

    WHY ICC..... 4

    BENEFITS IN NUMBERS..... 5

    REQUIREMENTS ..... 6

    COMMON CHALLENGES..... 7

**ESTABLISHING ICC..... 8**

**ICC AND ENTERPRISE ECOSYSTEM .....10**

    ROLES ..... 11

        1. *ICC Director*..... 12

        2. *ICC Manager*..... 14

        3. *Business Information Specialist*..... 16

        4. *Integration Architect* ..... 18

        5. *Functional Designer*..... 20

        6. *Technical Designer* ..... 22

        7. *Service Manager*..... 24

        8. *Test Manager* ..... 25

        9. *Security Architect* ..... 27

**DELIVERY MODELS.....29**

    ARTIFACTS DRIVEN MODEL ..... 30

    SPRINT MODEL ..... 31

**ICC ACTIVITIES .....32**

    ACTIVITIES FLOW ..... 33

    RECURRING EVENTS ..... 34

        1. *Development Meetings* ..... 35

        2. *Production Meetings* ..... 37

        3. *Disaster Recovery Test* ..... 39

    COMMUNICATION GAP ..... 40

**BEST PRACTICES.....42**

    CYCLIC DEPLOYMENT..... 43

    TESTING ..... 47

    TEST-DRIVEN DEVELOPMENT ..... 48

**ARTIFACT REVIEW .....50**

    SERVICE ORIENTED ARCHITECTURE ..... 51

**CONCLUSION .....52**

**APPENDIX - A: ICC CHECKLIST .....54**

**APPENDIX - A: PERSONS ASSIGNED TO ROLES .....57**

## Figures

<i>Figure 1: ICC Benefits</i>	5
<i>Figure 2: ICC stages and associated activities</i>	9
<i>Figure 3: ICC and Enterprise Ecosystem</i>	10
<i>Figure 4: ICC Roles</i>	11
<i>Figure 5: Activities Flow</i>	33
<i>Figure 6: Development meeting</i>	36
<i>Figure 7: Production meeting</i>	38
<i>Figure 8: Communication flow</i>	41
<i>Figure 9: Continuous Development process</i>	44
<i>Figure 10: Sequencing problem</i>	45
<i>Figure 11: Development and Deployment Cycle</i>	46
<i>Figure 12: Test Driven Development</i>	48

## INTRODUCTION

With 20 years of experience in business process integration, Friends Technology 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 Friends Technology 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
  - A. Service Oriented Architecture
  - B. Enterprise Service Bus
  - C. Messaging Models
  - D. Integration Test Methodology
4. Processes
  - A. Formalized steps for each process action
  - B. Document trace for each process
5. Standardization
  - A. Naming conventions, Common standards, Process standardization

Do not confuse ICC with a project or process model. 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
- Practically tested Disaster Recover
- Formal and active communication

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

## Why ICC

There are three 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.

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. ICC involves all stakeholders such as business people and main users in the continuous process development.

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.

## Benefits in numbers

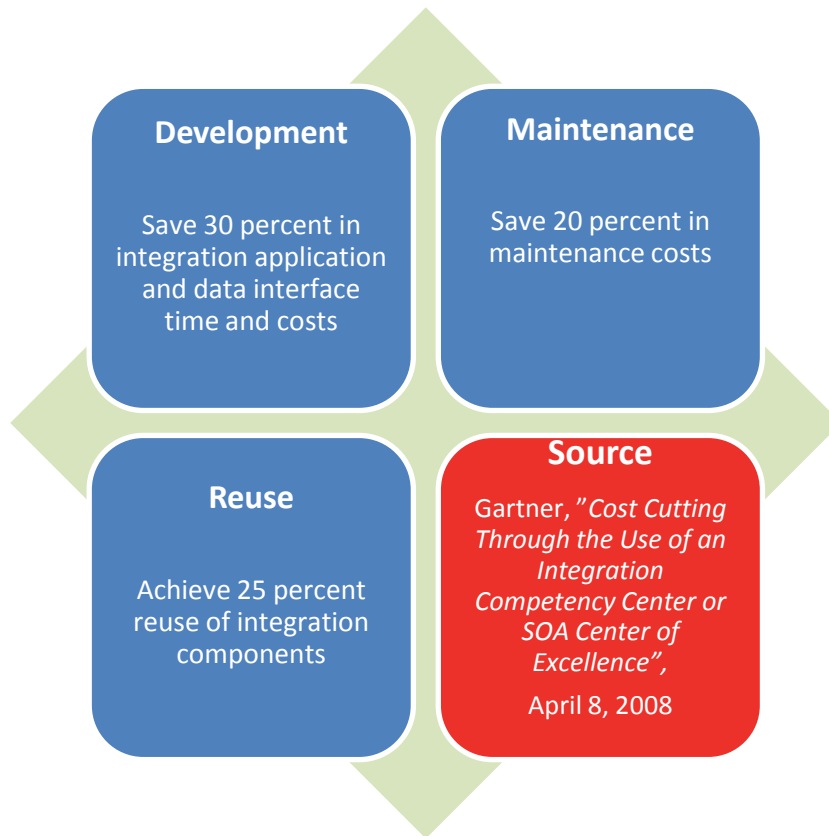


Figure 1: ICC Benefits

## Requirements

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

## Common Challenges

ICC as a concept is fairly simple. It is embodiment of the IT management best practices to deliver shared services. However, is an organizational concept: as every organization has a different DNA demanding specific personalization/customization effort to make the ICC initiative successful, it is 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.

*“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.

10. Cyclic Deployment

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

## 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.

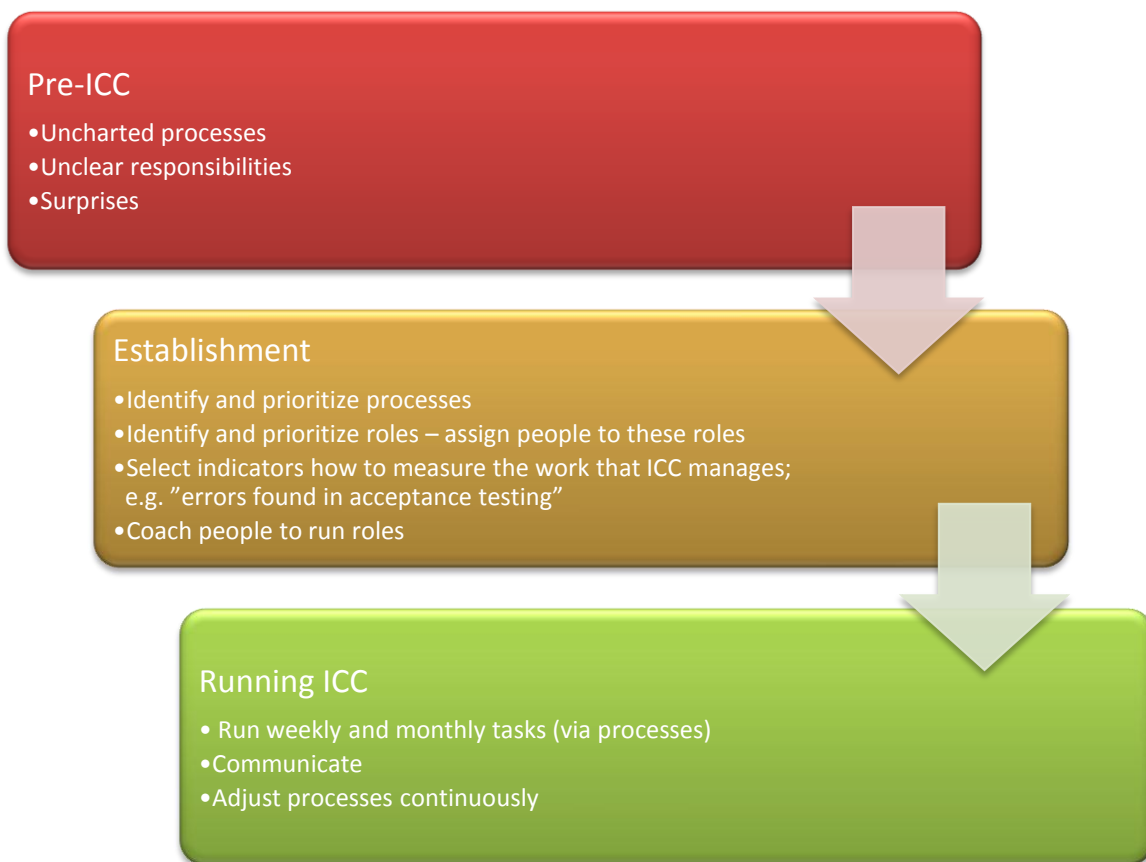
This approach has two main requirements:

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

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

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**

## ICC and Enterprise Ecosystem

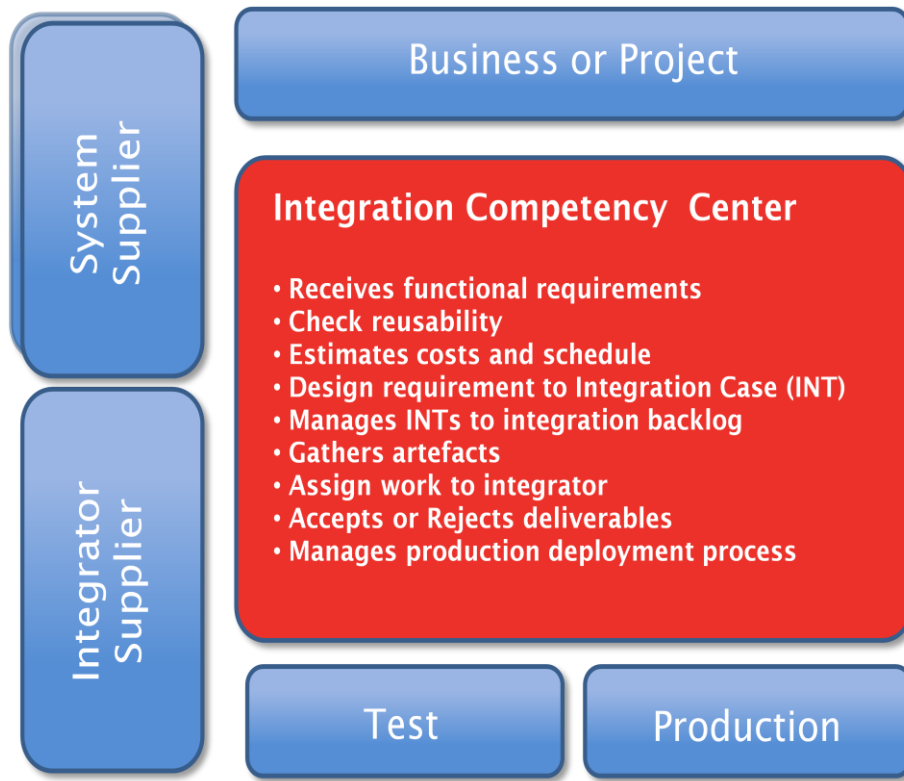


Figure 3: ICC and Enterprise Ecosystem

Integration work includes always several parties. The most common setup is described in Figure 2. The picture above illustrates the role of ICC – as 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.

## Roles

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

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.

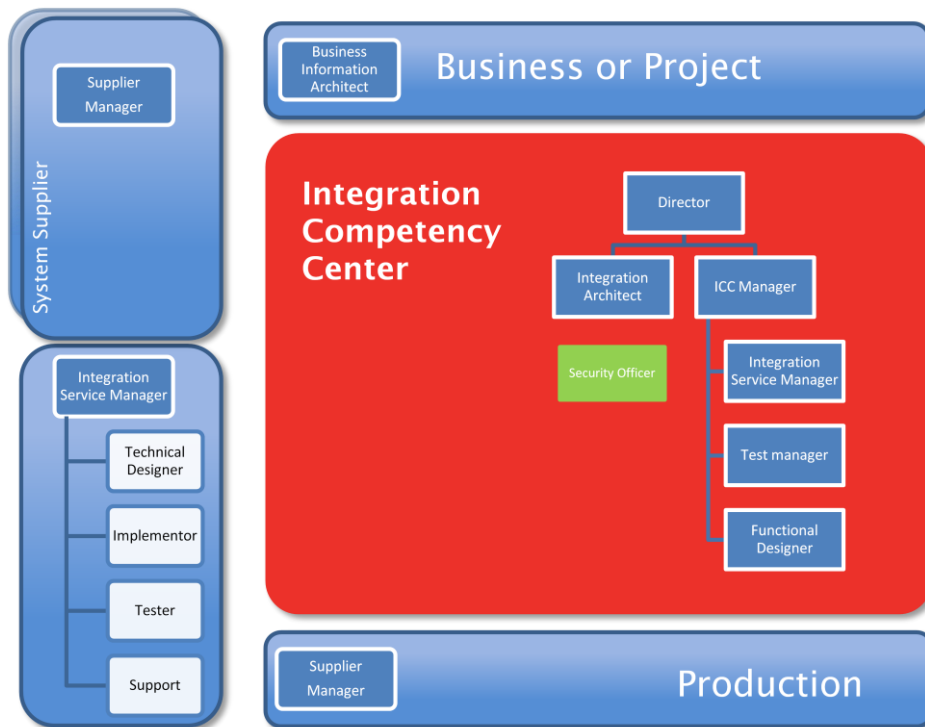


Figure 4: ICC Roles

## 1. ICC Director

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.

**Reports to:** Business Executive.

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
- Resolve prioritization and budget issues.

- Establish required systems and applications.
- 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 it's progress to the Steering/Management Group.
- Selects vendors of the hardware tools needed for integration efforts that may span servers, storage, and network items.

## 2. ICC Manager

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 serves as the right hand of ICC Director.

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

**NOTE:** This role is mandatory.

Specific activities vary depending on

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

The ICC Manager works with

- ICC Director
- Project teams
- 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 6 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

**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.

### 3. Business Information Specialist

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

**Reports to:** Business Unit Manager.

**NOTE:** Business Information Specialist role is optional.

Specific activities vary depending on

- Industry sector
- Business model

The Business Information Specialist works with

- Functional Designer
- Business architects
- Cross-functional management teams

Main activities are

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

## 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.
- 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. Integration Architect

Integration Architect is responsible of maintenance the overall integration architecture.

**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.

Specific activities vary depending on

- Industry sector
- Company size

The Integration Architect works with

- ICC Manager
- Business Information Specialists
- 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

**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.

## 5. Functional Designer

Functional Designer acts as a link between the “business need” and the technical solution. He or she should analyze the “business need” and find the artifacts required by Technical Designer and/or Implementer.

**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.

### The Functional Designer

- will specify “what is happening” in all Use-Cases.
- is responsible with the initial creation of all Use-cases with their elements i.e. Actors, Scenarios, Successful paths and Exceptions.
- will have to point out “what are the expected “Exceptions”.
- must define “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

#### 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.

## 6. Technical Designer

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.

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 ERP.

**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):** In agile models, Technical Designer is most likely the same person as Implementer. Furthermore, in smaller ICC organizations Integration Architect and Technical Designer are very likely to be carried out by a single 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):**

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

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.
- 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.

## 7. Service Manager

Depending on chosen project model this role is called Service Manager for Continuous Development or Project Manager for project methodologies.

**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:** 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 a 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

## 8. Test Manager

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

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.

**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.

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

- 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.

## 9. Security Architect

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

**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.

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.
- (Role based) authorization.
- Lead security incident management activities.
- Access control to the Integration solution and it's 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 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.
- 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.

## DELIVERY MODELS

In contrast with software development projects, integration projects requires a different approach in their development as well as deployment (End-game i.e. delivery). We have identified two possible delivery models, those are Artifacts Driven Model and Sprint Model.

## ***ARTIFACTS DRIVEN MODEL***

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

The model reflects the fact that artifacts are usually collected in an ad-hoc style.

Because of lack of integration competence in organizations - more often than not Integration suppliers are forced to use Artifacts Driven model.

Commonly, organizations do not understand what artifacts are needed for a given integration. This leads to a situation whereby organizations asks from the third parties one artifact at a time. In addition it's also common that organizations are not capable to evaluate the quality of artifacts.

The approach takes into account the fact that, in integration projects, very often artifacts are neither totally correct nor delivered in full. This reflects the complexity inherent in most integration projects.

The model 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. Therefore, it must be noted that a mature ICC makes this approach futile.

The model reflects the fact that artifacts are usually collected in an ad-hoc style.

Too often integration supplier is assigned to do an integration work with neither proper nor complete artifacts. With proper processes, a mature ICC should not have the need to use the Artifacts Driven model.

The Artifacts Driven model is the most commonly used approach although it's the most challenging for System Integrators.

In Artifacts Driven model, the integration work can only start after ICC has received all reviewed artifacts.

- 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 projec.
- 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.

## ***SPRINT MODEL***

The Sprint model, this approach is meant for bigger project, usually starting from 90+ days.

This approach has two main assumptions;

- Firstly, 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.
- Secondly, the model assumes that, business or projects will have to accept the fact that changes or new feature requests that are added during the current Sprint will not be delivered in the current Sprint but the next Sprint. normally the Sprint length is four weeks.

The “Sprint Model” is based on AIM i.e. Agile Integration Model. AIM is a project management methodology for agile solution development, it’s derived from SCRUM software development model.

AIM makes an effort to resolve some of the common failures of a typical integration project. Among the common failures are;

- The Chaos of Changes - usually project requirements do change drastically during the project lifecycle - from solution design to the moment of deployment.
- Unrealistic estimates of time, cost and quality of the solution - Project Managers as well as Developers tend to underestimate how much time and resource a project will require. The underestimation is usually due to the inaccurate requirement definitions and rapidly changing business needs - which in turn leads to requirement changes.

AIM gives the Supplier and other Stakeholders a possibility for the mutual understanding of the project status.

In addition, AIM brings forward the following aspects;

- AIM controls the delivery process of an Integration-Project from extracting the business needs (from business people) to the completion of a solution that fulfils the needs.
- AIM brings the stakeholders (e.g. customers, designers, developers) closer to the project execution and resolve problems that arise in multi-supplier projects.
- AIM does address the complications common in integration projects.

## ICC ACTIVITIES

To ensure that all tasks get done properly they must be controlled tightly. Here we enlist 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.

## ACTIVITIES FLOW

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

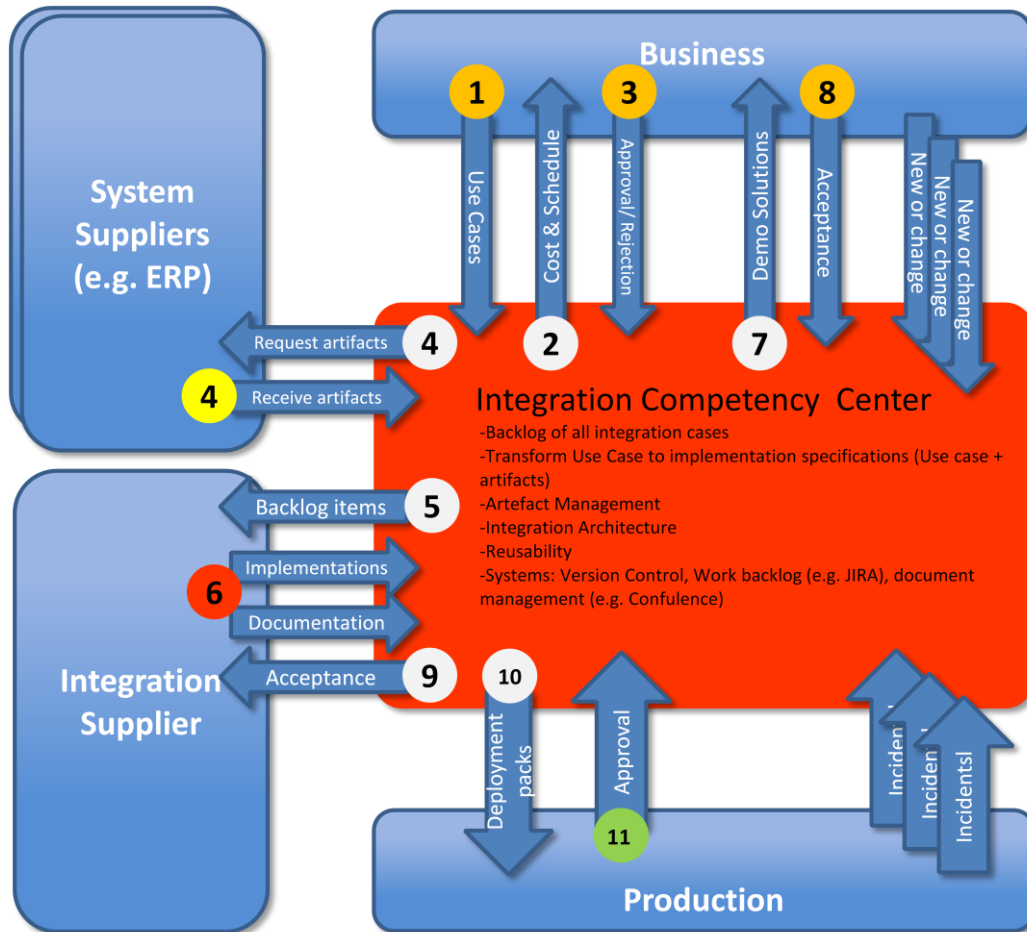


Figure 5: Activities Flow

## ***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.

## 1. Development Meetings

Development meetings should occur at least twice per month - preferably, as often as the pace of development.

Mandatory attendance:

- ICC Manager
- Test Manager (depending on deployment cycle length)
- Functional Designer(s)
- Project or Service Manager from Integration supplier

Optional attendance:

- All other roles

Weekly 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. Incidents from Production
  - A. Allocate resources for workaround ? Create a Bug or Task in backlog.
  - B. Allocate resources for problem solving ? Create Task in backlog.
7. Communicate what is going to be deployed in next deployment cycle (see page 42). 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 Director or ICC Manager.
8. 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	...	

9. Write a meeting memo

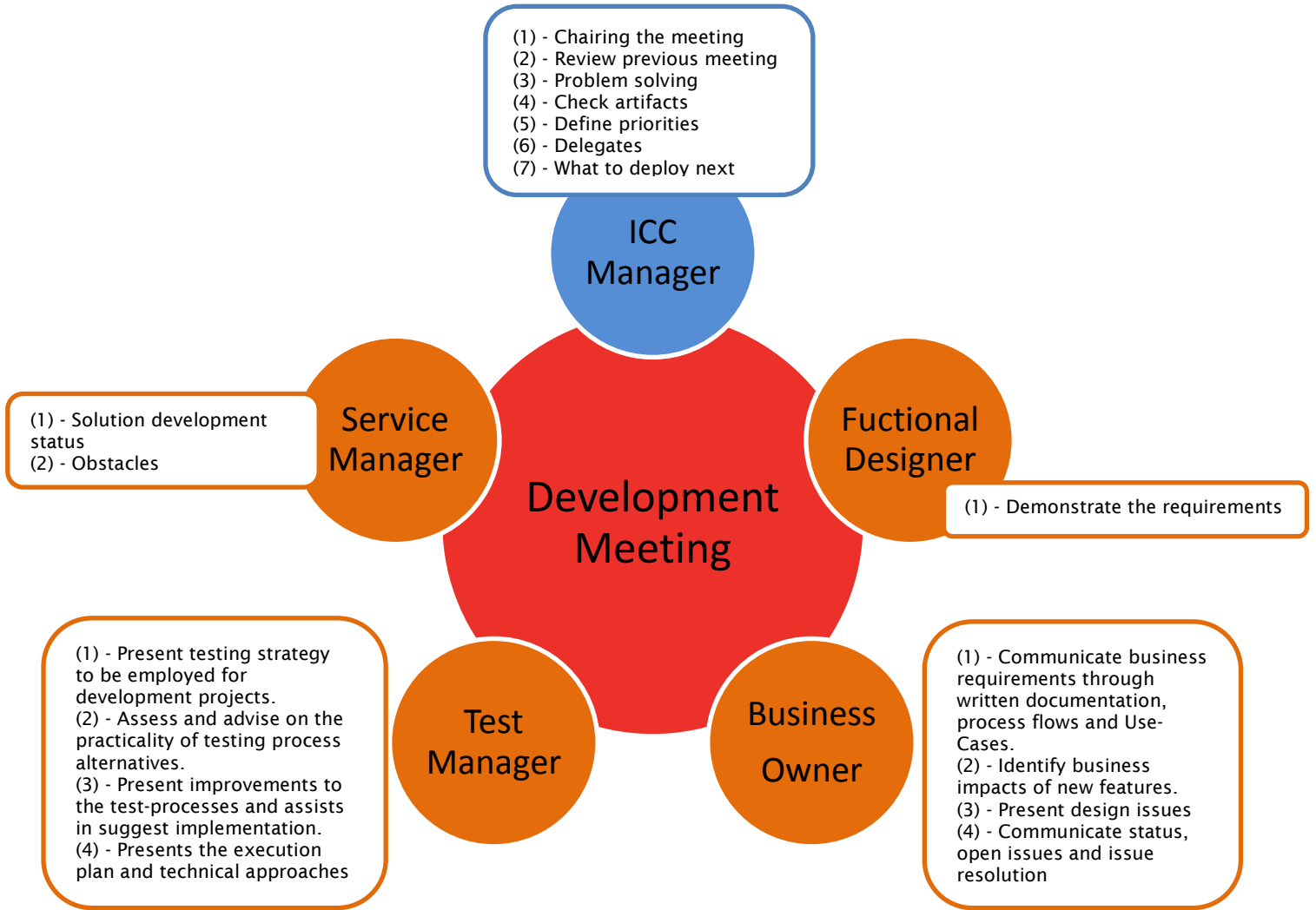


Figure 6: Development meeting

## 2. Production Meetings

Production meetings should occur every time when there are major production deployments.

Mandatory attendance:

- ICC Director
- ICC Manager
- Test Manager
- Production Service Manager
- Functional Designer(s)
- Business Information Specialist(s)
- Project Manager from the Integration supplier
- 

Optional attendance:

- All other roles

The meeting should cover and document following topics

1. Handle Production Report from previous production deployment
  - A. Problems
    - i. What should be done differently and how enhancements are achieved?
    - ii. How changes are communicated to stakeholders? and
    - iii. How changes to processes are going to be taken into action?
2. Adjust processes based on incidents, problems or business requirements. Weekly meetings provide list of problems or change request that might affect on processes.
3. Write a meeting memo

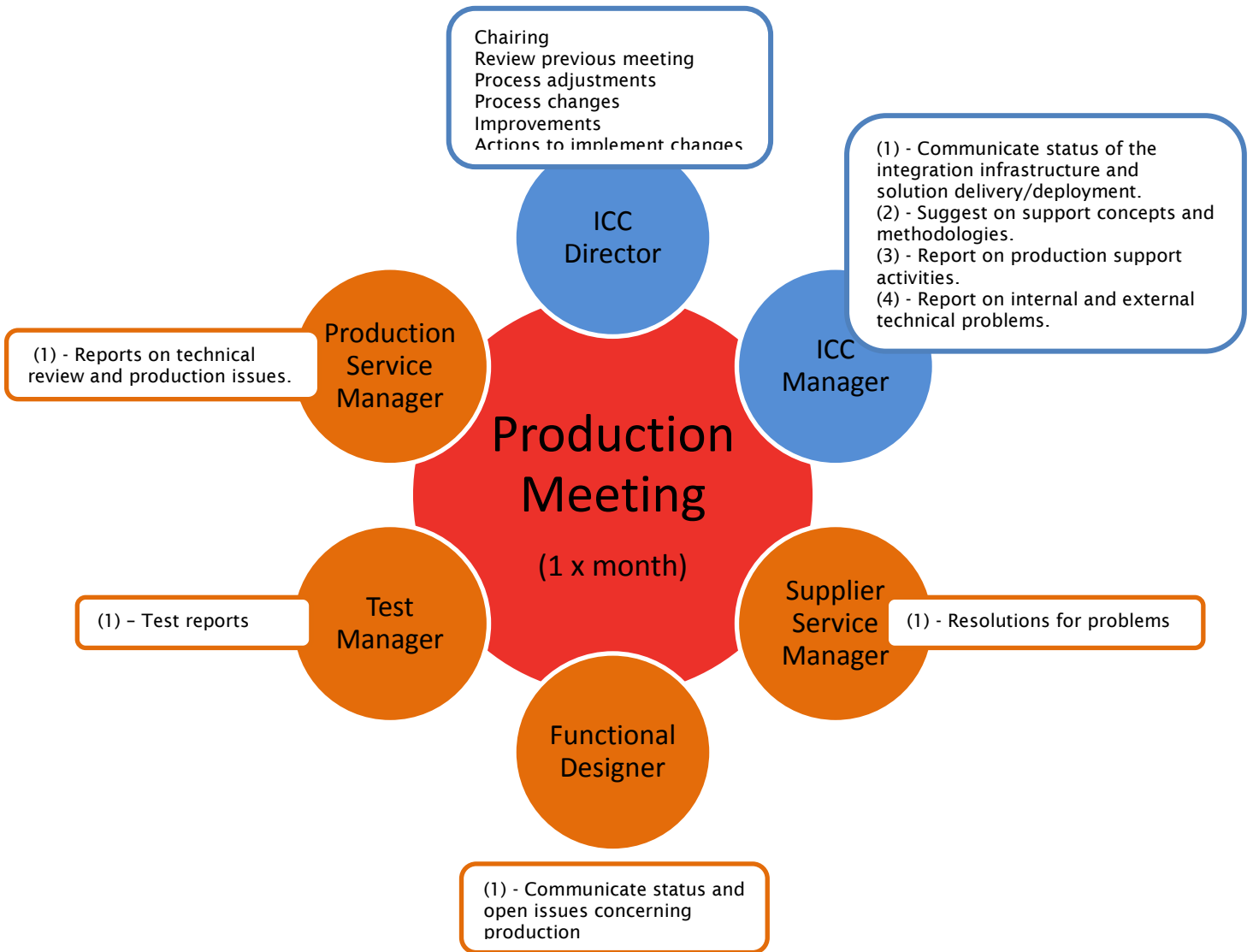


Figure 7: Production meeting

### 3. 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

## **COMMUNICATION GAP**

Imagine a situation where a business representative is arranging a business to business messaging between own organization and end customer. He or she must agree for example how end 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 a project, the deadline for delivery of solution is already set. The go-alive 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 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.

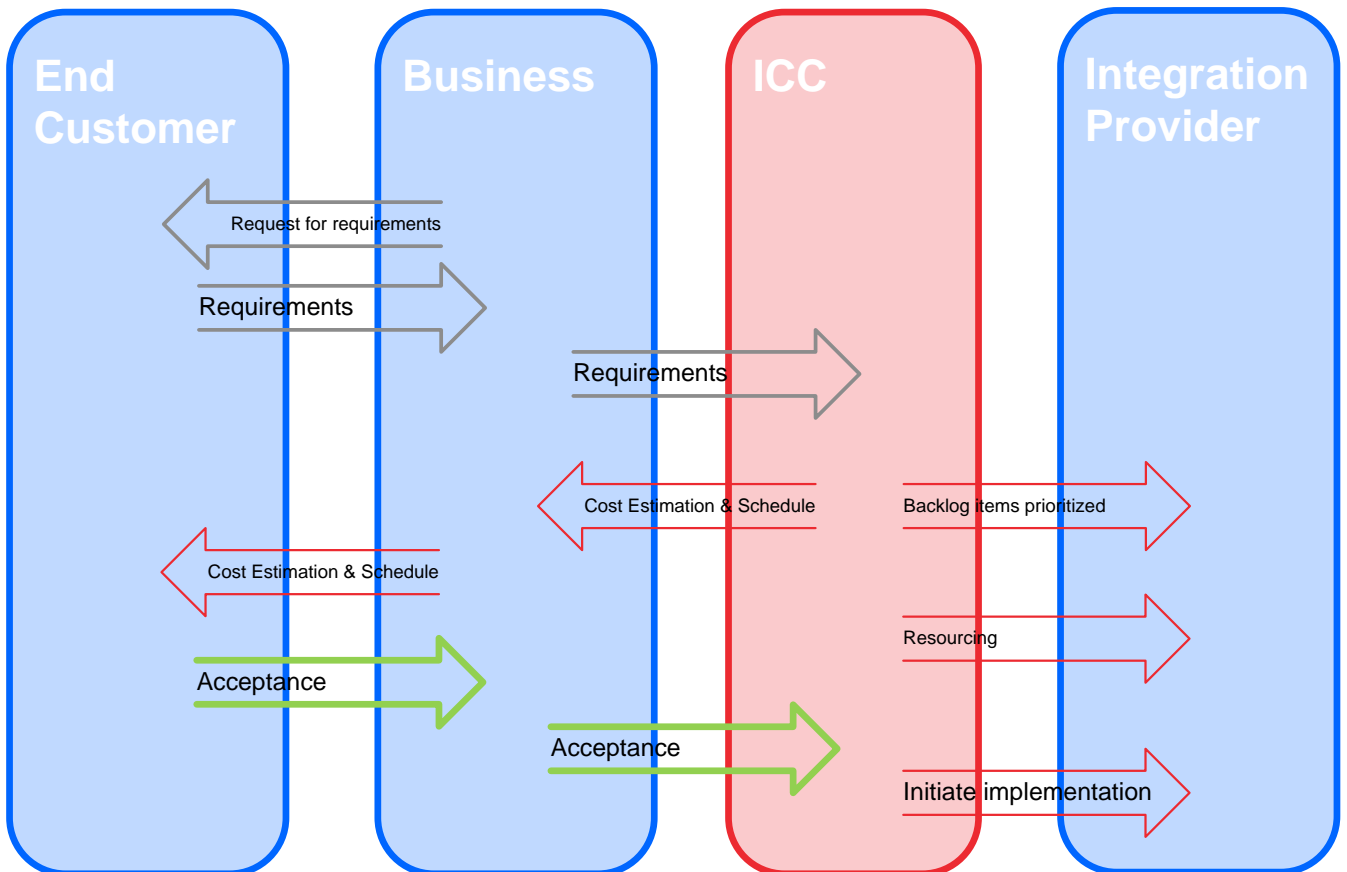


Figure 8: Communication flow

## 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; those are Cyclic Deployment, Testing, Test-Driven Development, Artifact Review and Service Oriented Architecture.

## ***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.

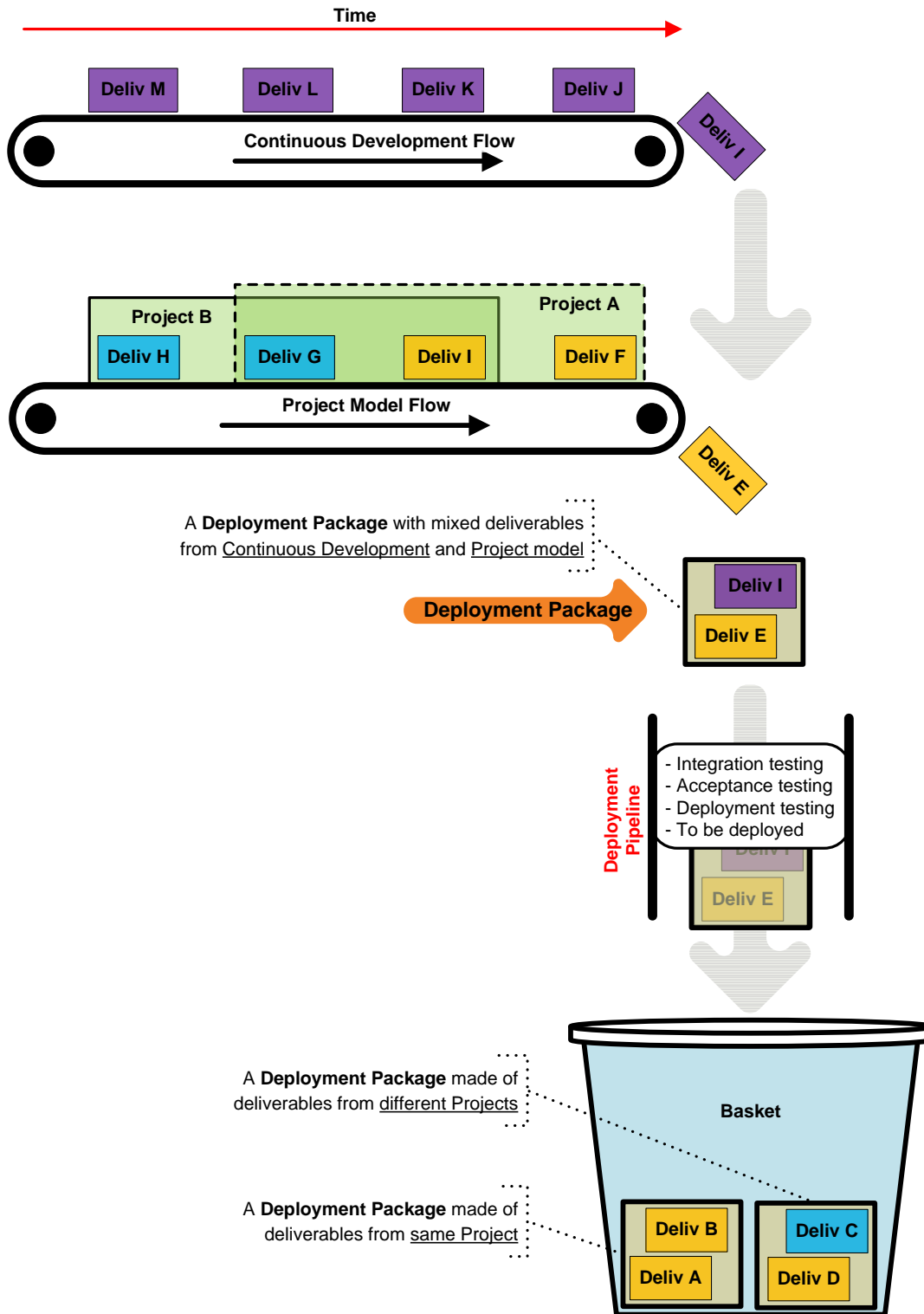


Figure 9: Continuous Development process

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.

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

Following picture illustrates sequencing problem.

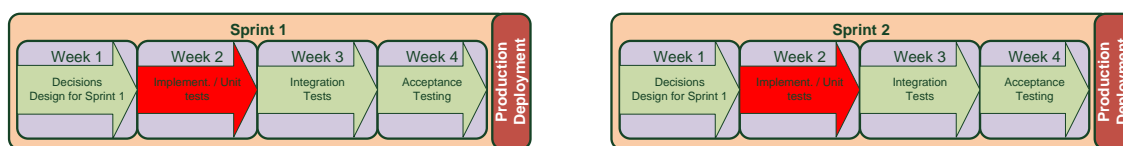


Figure 10: Sequencing problem

To ensure that integration supplier(s) work efficiently, ICC must separate deployment from development process.

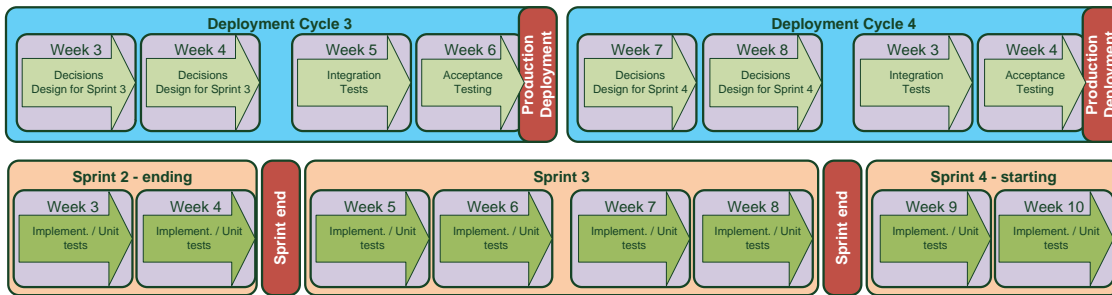


Figure 11: Development and Deployment Cycle

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.

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.

## TESTING

It is easy to agree that testing is a best practice - a mandatory part of multiple phases of development (*unit testing* and *system testing*) and deployment (*integration testing*, *acceptance testing* and *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.
- Test cases MUST be derived from Use Cases.
- 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 should correspond to real-world scenarios.
- 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.

## 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.

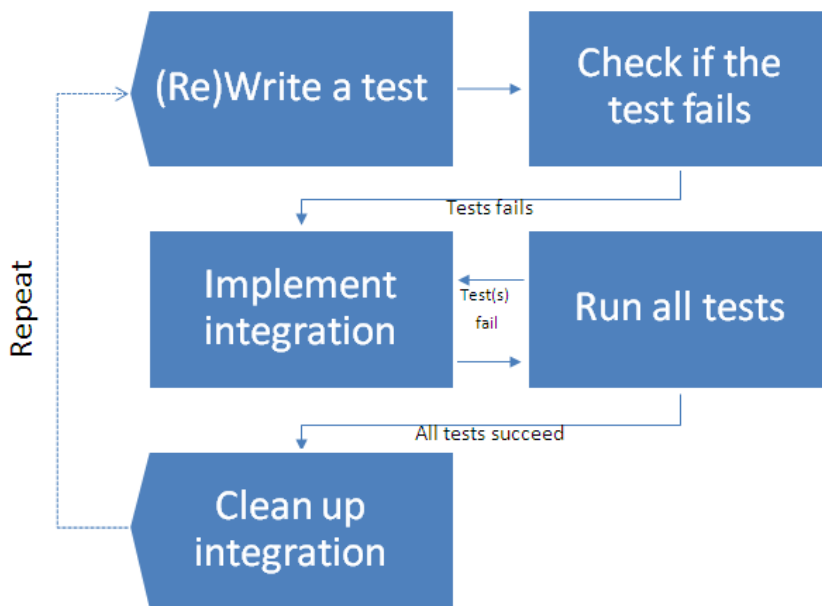


Figure 12: Test Driven Development

When considering TDD, pay attention to following issues

- TDD requires test automation software compatible with the [Integration Platform](#).
- TDD requires training for Designer who is defining the automated test cases and for the developers who uses test automation software.
- 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.
- 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. 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.

## ARTIFACT 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:

- Require Interface test report of third party interfaces.
- Require Test messages that were used in 3rd party interface test.
- Identify 3rd party interfaces that are risky. Following reasons increase interface's risk level
  - It is created for this need. In other words, it is used for the first time.
  - 3rd party documentation, schema or other artifact is missing.
  - 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 a end-element.
- Design 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.
- Require full documentation of external interfaces.
  - Integration supplier needs all the information of interface that is used to pull out or push in information.
- 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.

## ***SERVICE ORIENTED ARCHITECTURE***

This chapter covers the third aspect of Best Practises for ICC group for managing the Service Oriented Architecture implementation. The common reason for SOA project to fail is simply the lack of proper skills. Therefore ICC must have resources to build the SOA Centre of Excellence. Remember that SOA is definitely NOT the IT solve-all-problems approach.

Following rules answer the questions “When to use SOA?” and “When it should be avoided.”

SOA approach may not be the best choice if your solution falls in one or more of the following areas:

- Event-driven applications
  - Securities trading
  - Telecom services activation
  - Real-time fraud detection
  - Real-time supply/demand chain
  - Airline operations tracking
- Near real time data consistency
- Batch-Oriented Processing
  - Bill/statement printing
  - Data warehouse loading
  - Database reconciliation
- Monitoring Applications
  - System and Network Monitoring
  - Indoor Positioning
  - Industrial Processes
- EDI-Style B2B

SOA fits best to request-reply messaging when

- Reusability between data consumers is a major goal
  - Composite Service “getCustomerData” is using multiple component services to get a single-view to Customer as a whole concept
- Requirement for reply times are more than 1 second (semi-latency services).
- Data must be published as a “single-view” service.
- Near-Real-Time Business to Business messaging
- Underlying layer in Self-Service Portals
  - Supplier portal
  - Customer portal
  - Employee portal
  - Citizens (in e-government)
- Multichannel Systems
  - Web shop uses same underlying “receiveSalesOrder” service as other systems

## CONCLUSION

As you have seen, we have outlined our experience in establishing ICC Group. In the first chapter we gave a brief account to the reasons, requirements and the challenges of establishing an ICC Organization.

In the second chapter we went through the basic steps to establishing an ICC organization – Establishing the ICC, Defining and choosing the processes, Establishing communication procedures as well as Systems.

Chapter three 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 lead 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 four introduces FREnds delivery models. Those are Artifacts Driven Model and Sprint Model. These models reflect our experience and common challenges in integration projects.

In chapter five 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 six introduces the Best Practices through several of the central activities in FREnds 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 FREnds 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 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.

Therefore, ICC may be defined as a management and co-ordination group for a set of related projects and activities that collectively do share some similarities that may lead to the development of key knowledge in a given business area. ICC provides the direction, control, optimization, consistency and long-term focus for these projects.

ICC breaks solution delivery down into manageable components (i.e. projects, deployment package and change request) with monitoring and review points with aim to build a knowledge base for reusability purpose.

By coordinating projects under ICC with an overall vision of the desired results - a broader knowledge of the business is established. This provides a means for sharing and re-using of key components among projects.

While the size and organizational structure of any given company may or may not have the roles as defined in this paper, we in FREnds do think that - the key thing is for the company to establish the ICC Group in order to take the responsibility of the integration efforts.

## APPENDIX - A: ICC CHECKLIST

<b>COMPANY LEADRESHIP/MANAGEMENT</b>	
Is ICC initiatives mapped onto Company's objectives?	
Have responsible management approved the ICC establishment?	

<b>ICC - GROUP</b>	
Determine if the ICC Group is established, i.e. ICC Manger, Information Specialist, Service Manager and Integration Architect.	
Are all members of ICC Group aware of ICC commitment?	
Do all ICC members understand the processes concerning ICC activities?	
Do ICC members understand the intentions and expectations of ICC Group?	
Does everyone involved in ICC Group understand their level of involvement, roles, 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 demonstratable and measured improvements towards excellence with respect to ICC?	

<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 ICC Group 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?	
What ensures that the source code changed is the most recent version of the program?	

How are Business Owner involved in ICC activities?	
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**

Is there a systematic communication <i>within</i> and <i>around</i> ICC Group?	
Is communication between ICC Mangers, Staff and Customer established and maintained?	
Is there active engagement with all Stakeholders/Partners <i>within</i> and <i>outside</i> ICC Group?	
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**

Examine the open issues list to ensure all significant issues have been addressed	
Lessons learned from project reviews are shared and adopted within the Company?	

## APPENDIX - A: PERSONS ASSIGNED TO 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					