Unified Process

Iterative and Incremental

Rolf-Helge Pfeiffer - Anders Kalhauge

cphbusiness
COPENHAGEN BUSINESS ACADEMY

Fall 2016

Agenda

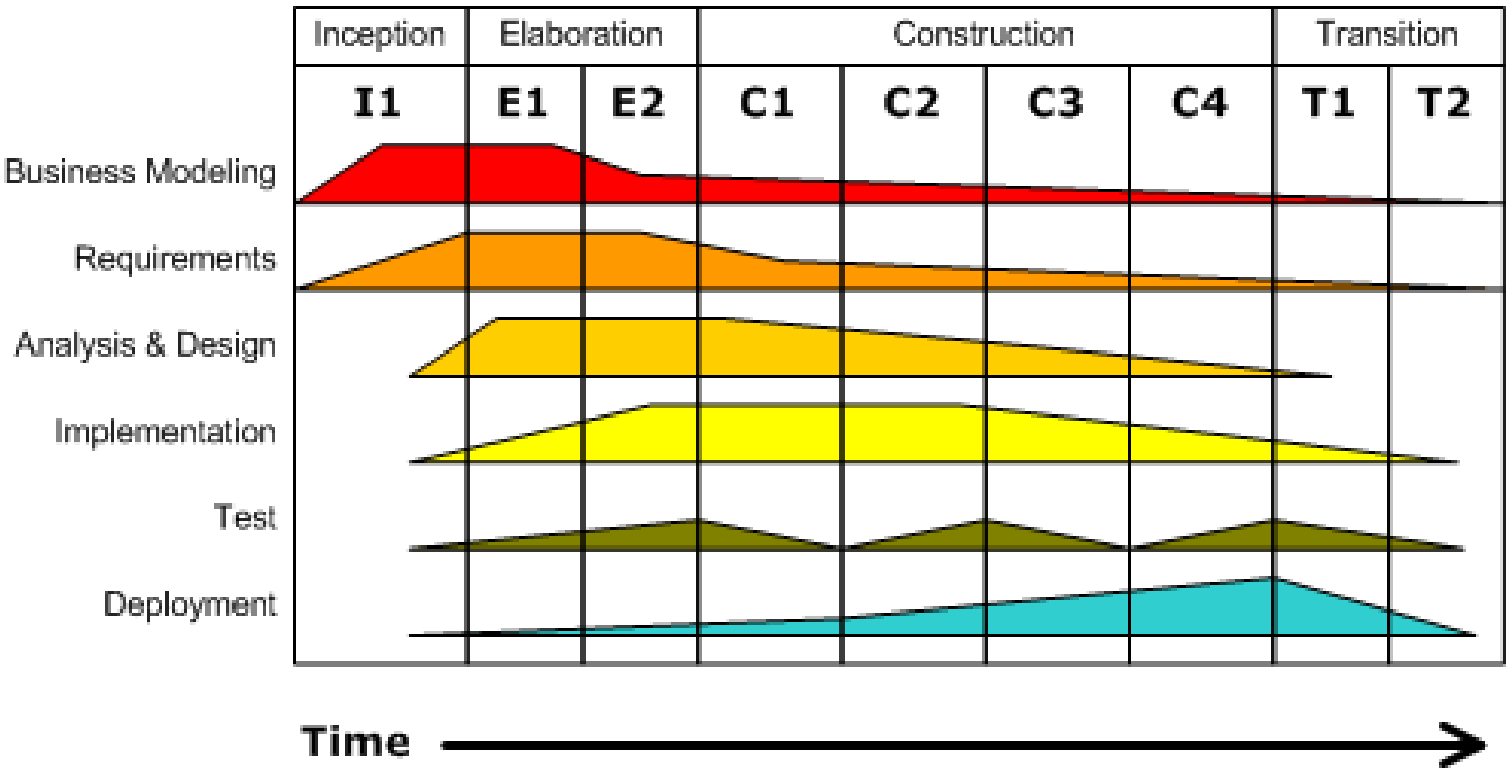
- Unified Process overview
- Iterations

The Unified Process is

- Use-Case Driven
- Architecture-Centric
- **Iterative and Incremental**

Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



- Business modeling
- Requirements Capture: From Vision to Requirements
- Capturing the Requirements as Use Cases
- Analysis
- Design
- Implementation
- Test

The Phases

- Inception
  Launches the Project
- Elaboration
  Makes the Architectural Baseline
- Construction
  Leads to Initial Operational Capability
- Transistion
  Completes Product Release

The goal in the inception phase is to make the business case to the extend necessary to justify lauching the project.

- Cost/benefit
  Will the gains accruing from the use or sale of the software product more than offset the cost of developing it?
- Time to market
  Will it reach the market (or internal application) in time to obtain these gains?
- Resolve system scope
- Resolve ambiguities in the requirements needed in this phase
- Establish a candidate architecture
- Mitigate the critical risks

- A feature list.
- A first version of a business (or domain) model that describes the context of the system.
- A first cut of the models representing a first version of the use-case model, the analysis model, the design model. Of the implementation model and test model, there may be something rudimentary. There is also a first version of the supplementary requirements.
- A first draft of a candidate architecture description with outlines of views of the use case, analysis, design, and implementation models.
- Possibly a proof-of-concept exploratory prototype, demonstrating the use of the new system.
- An initial risk list and a use-case ranking list.
- The beginnings of a plan for the entire project, including a general plan for the phases.
- A first draft of the business case, which includes: business context and success criteria (revenue projection, market recognition, project estimate).

The goal in the elaboration phase is to capture most of the remaining requirements, formulating the functional requirements as use cases.
Also a sound architectural foundation - the architectural baseline - must be established.

- Monitor remaining risks
- Fill in further details of the project plan.
- Judge the Worth of the Business Case

- Preferably a complete business (or domain) model which describes the context of the system.
- A new version of all models: use cases, analysis, design, deployment, and implementation. (At the end of the elaboration phase these models will be complete to less than 10% apart from the use case and analysis model that may include more (in some cases up to 80%) use cases to ascertain that the requirements have been understood. The majority of all use cases have been understood to make sure that no architecturally important use cases are left aside and that we can estimate the costs of introducing them.)
- An executable architectural baseline.
- An architecture description, including views of the use case, analysis, design, deployment, and implementation models.
- Updated risk list.
- Project plan for the construction and transition phases.
- A preliminary user manual (optional).
- Completed business case, including business bid.

The team working in the construction phase, starting from an executable architecture baseline and working through a series of iterations and increments, develops a software product ready for initial operation in the user environment.

- Project plan for the transition phase.
- The executable software itself—the initial-operational-capability release. This is the final build from construction.
- All artifacts, including models of the system.
- Maintained and minimally updated architecture description.
- Preliminary user manual in enough detail to guide beta users.
- Business case, reflecting situation at end of phase.

This phase focuses on establishing the product in the operational environment.

- Find out whether the systme really does what the business and its users request.
- Discover unanticipated risks.
- Note unresolved problems.
- Find failures.
- Fix ambiguities and gaps in the user documentation.
- Focus on areas where users appear to be deficient and in need of information or training.

- The executable software itself, including installation software.
- Legal documents such as contracts, license documents, waivers, and warranties.
- Completed and corrected product release baseline including all models of the system.
- Completed and updated architecture description.
- Final user, operator, and system administrator manuals and training materials.
- Customer support references and web references on where to find more information, to report defects, and to find information on fixes and upgrades.

Iterations in UP are

- Mini-projects
- Controlled
- Planned
- Minimize unforseen problems

Iterations

- deals with a group of use cases that together **extend** the usability of the product developed so far.
  - Additive increments - more code, more features
  - Qualitative increments - better code
- deals with the most important risks
  - Minimize unforseen problems

Controlled iterations

- reduces the cost risk to the expenditures on a single increment.
- reduces the risk of not getting the product to market on the planned schedule.
- speeds up the development tempo.
- makes it easier to adapt to changing requirements.

A medium sized Danish municipality runs two ferry lines which are serviced by three permanent ferries. The ferry lines service two small islands, one close by (approx. 20 minutes) and one about an hour away. The biggest ferry sails mainly on the long route, but can on rare occasions be used on the short one. A medium sized ferry is used on the short route in summer, and on both rou-tes in winter. The smallest ferry is used only on the short route. When one or more ferries are in dock, one of two substitute ferries might be borrowed from a neighbour municipality.

The municipality wants a system for reservations. The largest ferry has a movable deck, which when lowered supports more private cars but fewer lorries. The medium sized ferry can be rearranged to support big machinery as harvesters and big tractors. This rearrangement is not simple, and has to be planned at least 24 hours ahead. The smallest ferry can only transport people, no vehicles. On the short route only residents can bring cars, because the island is quite small and is subject to nature conservation.

The municipality already has a reservation system, but it is not user friendly, and has shown to be prone to changes; due to the purchase of a new medium sized ferry; the system has not been working for the short line for the last couple of months. The municipality is negotiating with the neighbour municipality about a merger between their shipping activities. Ticket prices differ a lot between the two municipalities, and even between the different ferry lines.

At some lines to the minor islands, citizens with permanent addresses on the islands are entitled free rides, rules for how many changes over time and from line to line. Offering online tickets has been discussed. For inspiration concerning content of reservations for ferry lines you can see links such as:

- http://www.scandlines.com/
- http://www.aferry.co.uk/
- http://www.kalundborg.dk/Forside-2.aspx

Configuration examples