Chapter 3
Prescriptive Process Models

Moonzoo Kim
CS Division of EECS Dept.
KAIST

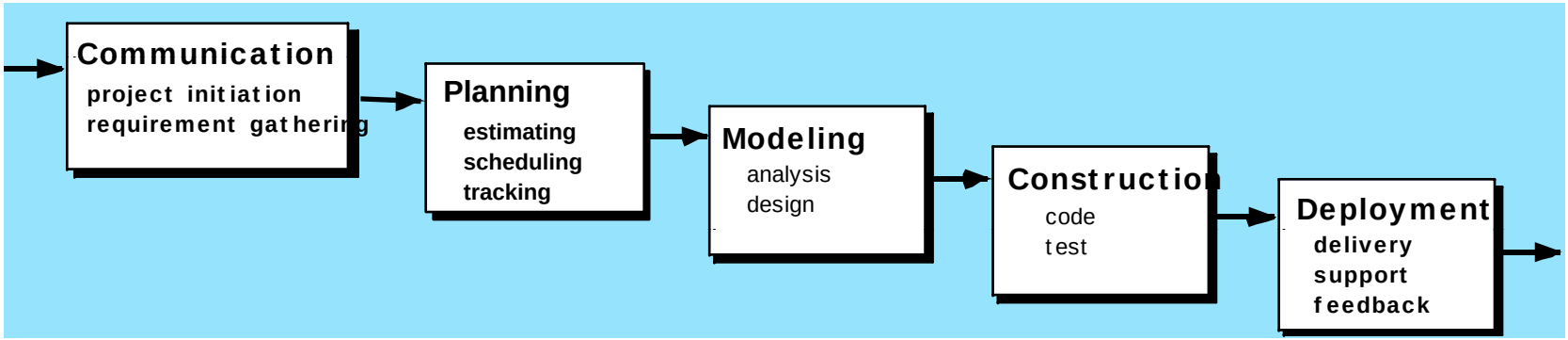
Prescriptive Models

- Prescriptive process models advocate an *orderly* approach to software engineering

That leads to a few questions …

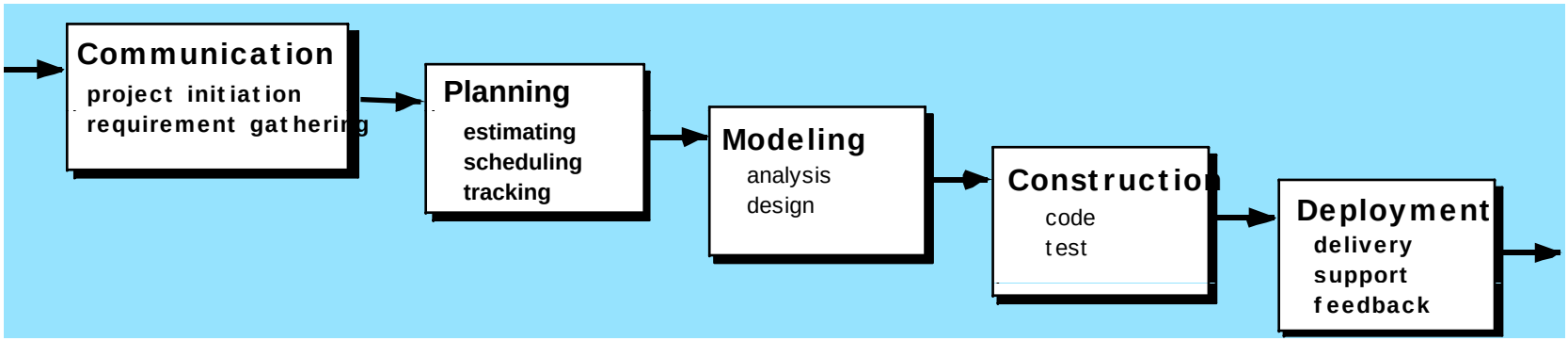
- If prescriptive process models strive for structure and order, are they inappropriate for a software world that thrives on change?
- Yet, if we reject traditional process models (and the order they imply) and replace them with something less structured, do we make it impossible to achieve coordination and coherence in software work?

The Waterfall Model (1/2)



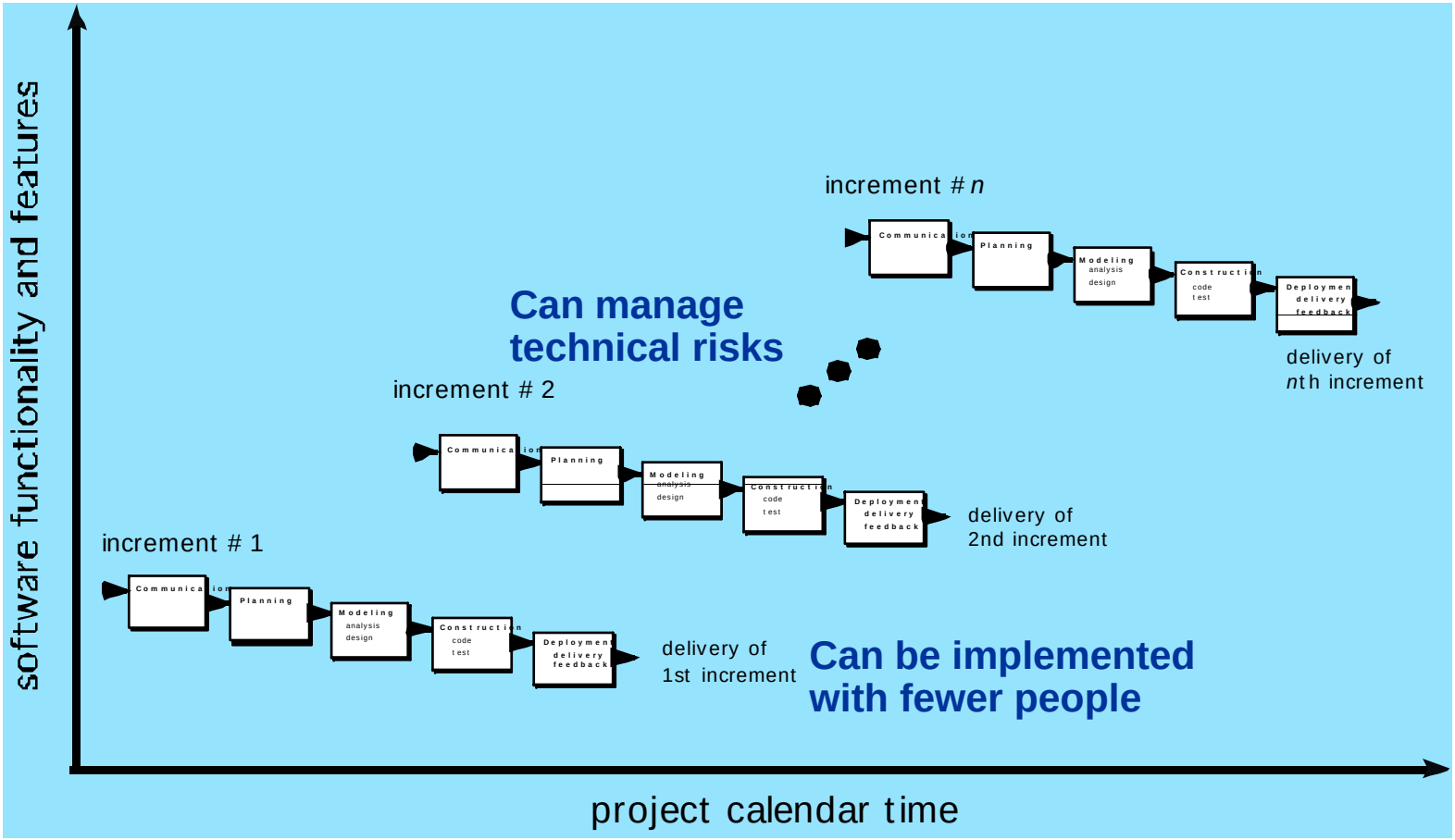
- Which problems does the waterfall model have?
  1. Real projects rarely follow the sequential flow
  2. Difficult to accommodating the uncertainty in requirements
  3. A working version of SW will not be available until late in the project

The Waterfall Model (2/2)

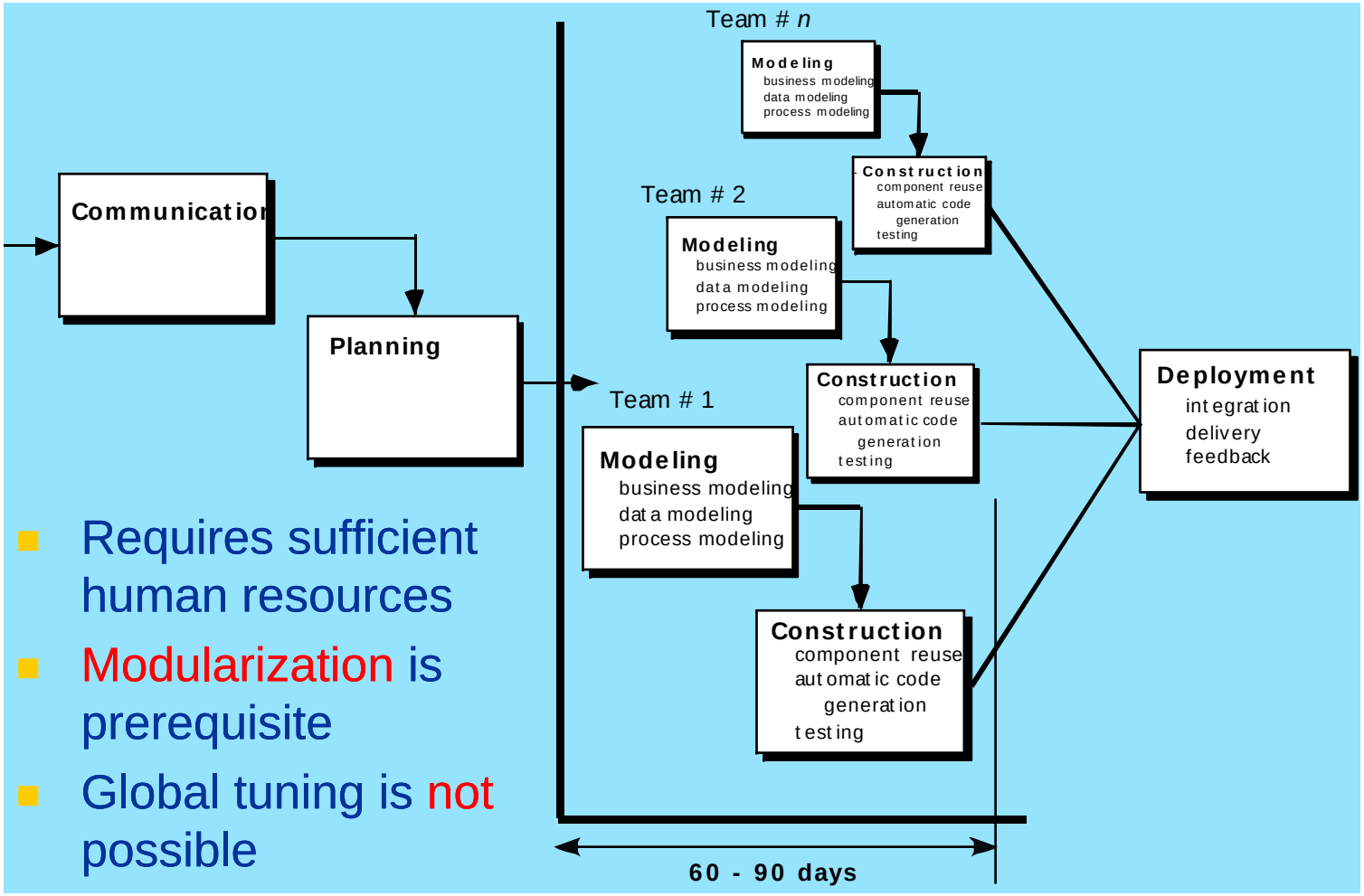


- Bradac[BRA94] found that the linear nature of the waterfall model leads to “blocking states”
  - Where some members must wait for other members of the team to complete dependent tasks
  - Especially, at the beginning of the project
- Still, however, the waterfall model serve as a useful process model where requirements are fixed and work is to proceed to completion in a linear manner

The Incremental Model



The RAD (Rapid Application Development) Model



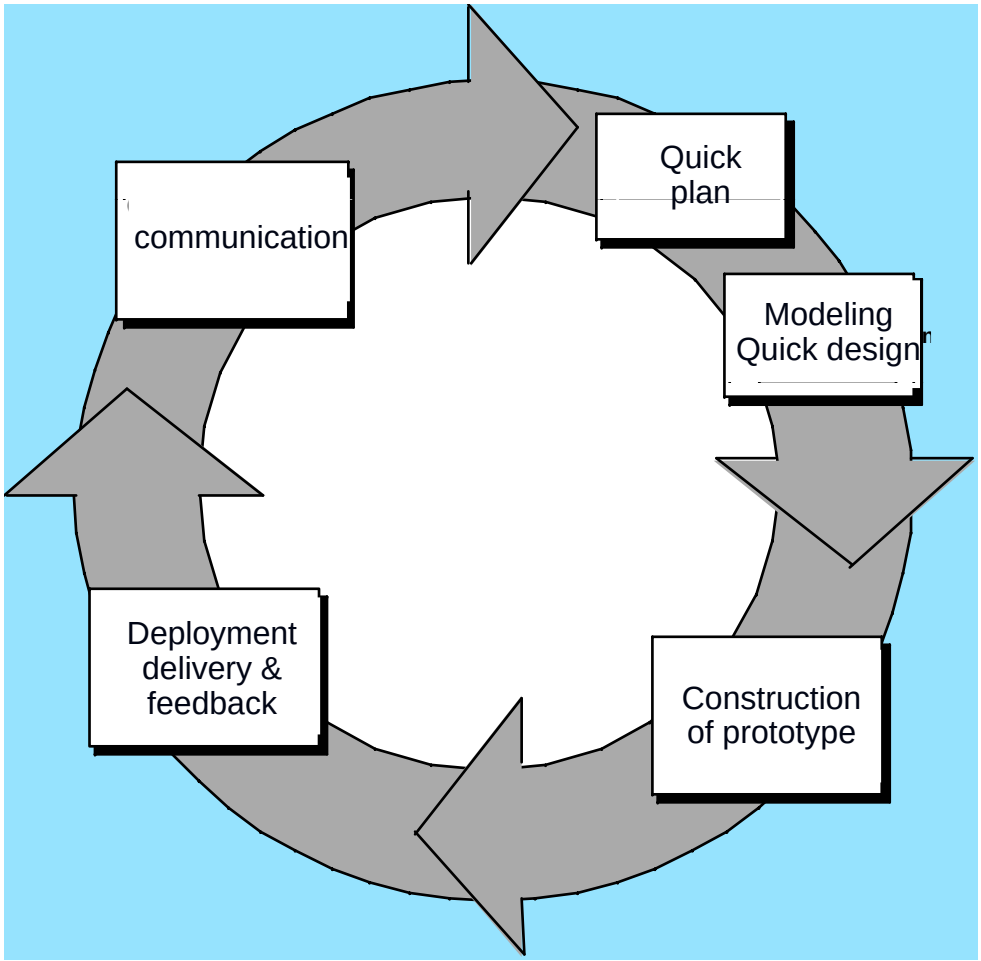
- Requires sufficient human resources
- Modularization is prerequisite
- Global tuning is not possible

Evolutionary Models: Prototyping (1/2)

- A prototyping paradigm is the best-fit for the following situations
  - A customer does not identify detailed requirements for SW
  - SW engineers are not sure of the efficiency of an algorithm, usability of SW, and so on.
- In other words, prototyping paradigm helps SW engineers and the customers to understand what is to be built
- The quick design and implementation focuses on a representation of those aspects of the SW that will be visible to the customer
  - Ideally, the prototype serves as a mechanism for identifying SW requirements

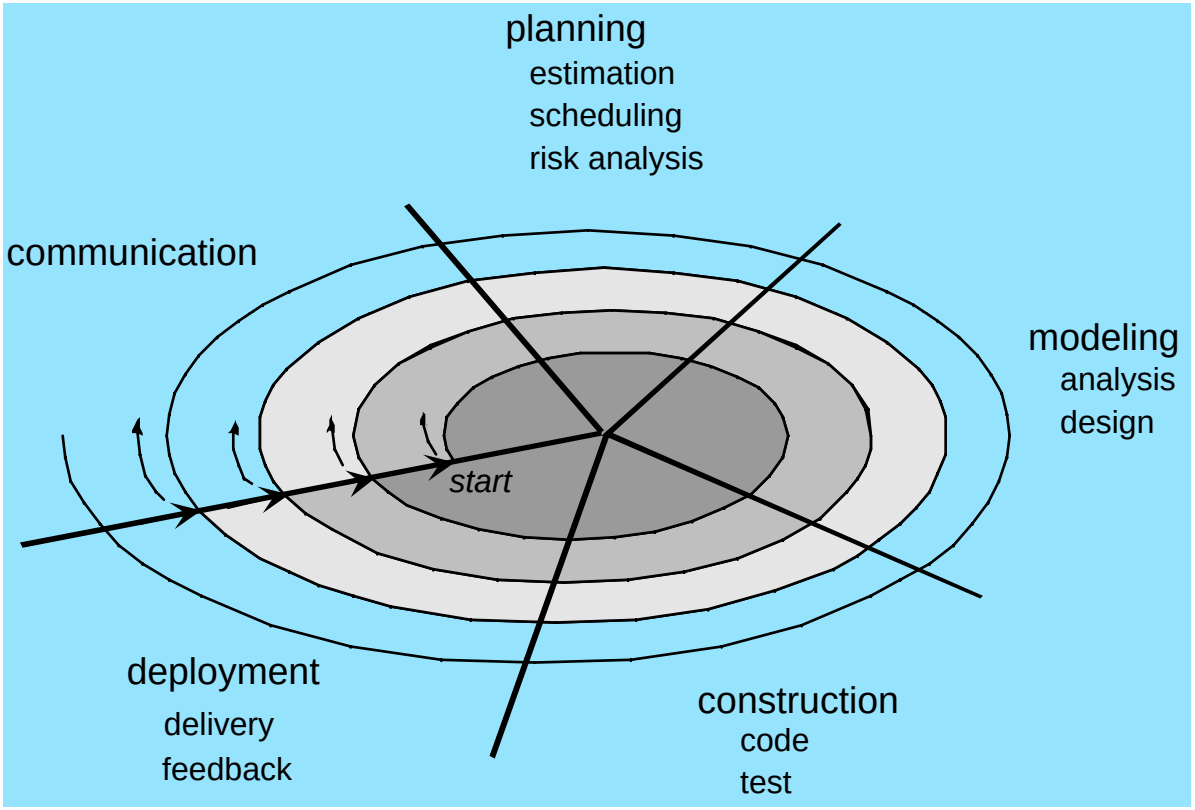
KAIST

Evolutionary Models: Prototyping (2/2)



- Some problems in prototyping paradigm
  - SW engineers try to modify the prototype to use as a working version
  - Once the customer see the working prototype, he/she expects to get working product soon

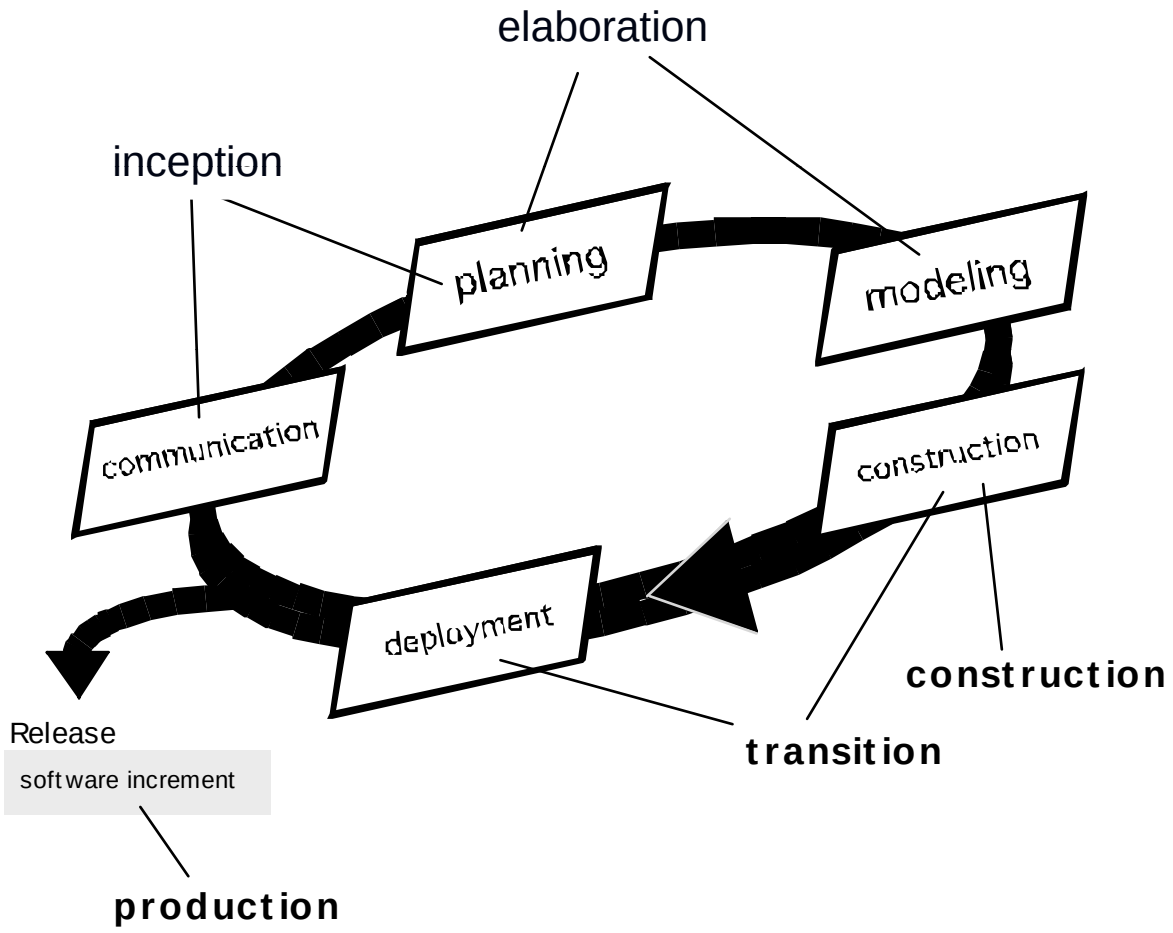
Evolutionary Models: The Spiral



Still Other Process Models

- Component based development—the process to apply when reuse is a development objective
- Formal methods—emphasizes the mathematical specification of requirements
- AOSD—provides a process and methodological approach for defining, specifying, designing, and constructing *aspects*
- Unified Process—a "use-case driven, architecture-centric, iterative and incremental" software process closely aligned with the Unified Modeling Language (UML)

The Unified Process (UP)



UP Work Products

Inception phase

Vision document
Initial use-case model
Initial project glossary
Initial business case
Initial risk assessment.
Project plan,
  phases and iterations.
Business model,
  if necessary.
One or more prototypes

Elaboration phase

Use-case model
Supplementary requirements
  including non-functional
Analysis model
Software architecture
  Description.
Executable architectural
  prototype.
Preliminary design model
Revised risk list
Project plan including
  iteration plan
  adapted workflows
  milestones
  technical work products
Preliminary user manual

Construction phase

Design model
Software components
Integrated software
  increment
Test plan and procedure
Test cases
Support documentation
  user manuals
  installation manuals
  description of current
    increment

Transition phase

Delivered software increment
Beta test reports
General user feedback

Quick Overview of SafeHome

- The SafeHome company has developed an innovative HW box that implements wireless Internet (802.11) connectivity in a very small form factor (the size of a matchbook).
- The idea is to use this technology to develop and market a comprehensive home automation product line.
  - This would provide
    - security functions
    - control over telephone answering machines
    - lights
    - heating
    - air conditioning
    - home entertainment devices.
- The first generation of the system will only focus on home security since that is a market the public readily understands.

KAIST

Selecting a Process Model, Part 1(pg 85-86)

- **The scene:**
  - Meeting room for the software engineering group at CPI Corporation, a (fictional) company that makes consumer products for home and commercial use.
- **The players:**
  - Lee Warren, engineering manager;
  - Doug Miller, software engineering manager;
  - Jamie Lazar, software team member;
  - Vinod Raman, software team member;
  - Ed Robbins, software team member.
- **The conversation:**
- **Lee:** So let's recapitulate. I've spent some time discussing the *SafeHome* product line as we see it at the moment. No doubt, we've got a lot of work to do to simply define the thing, but I'd like you guys to begin thinking about how you're going to approach the software part of this project.

- **Doug:** Seems like we've been pretty disorganized in our approach to software in the past.
- **Ed:** I don't know, Doug. We always got product out the door.

Doug: True, but not without a lot of grief, and this project looks like it's bigger and more complex than anything we've done in the past.

- **Jamie:** Doesn't look that hard, but I agree ... our ad hoc approach to past projects won't work here, particularly if we have a very tight timeline.
- **Doug (smiling):** I want to be a bit more professional in our approach. I went to a short course last week and learned a lot about software engineering ... good stuff. We need a process here.

KAIST

- **Jamie (with a frown):** My job is to build computer programs, not push paper around
- **Doug:** Give it a chance before you go negative on me. Here's what I mean. [Doug proceeds to describe the process framework described in Chapter 2 and the prescriptive process models presented to this point.
- **Doug:** So anyway, it seems to me that a linear model is not for us ... assumes we have all requirements up front and knowing this place, that's not likely.
- **Vinod:** Yeah, and that RAD model sounds way too IT- oriented ... probably good for building an inventory control system or something, but it's just not right for *SafeHome*
- **Doug:** I agree.
- **Ed:** That prototyping approach seems OK. A lot like what we do here anyway.
- **Vinod:** That's a problem. I'm worried that it doesn't provide us with enough structure.
- **Doug:** Not to worry. We've got plenty of other options, and I want you guys to pick what's best for the team and best for the project.

Selecting a Process Model, Part 2 (pg90-91)

- The players:
  - Lee Warren: engineering manager
  - Doug Miller: SE manager
  - Ed and Vinod: members of the SE team

The conversation: (Doug describes evolutionary process options)

- Ed: Now I see something I like. An incremental approach makes sense and I really like the flow of that spiral model thing. That's keeping it real.
- Vinod: I agree. We deliver an increment, learn from customer feedback, replan, and then deliver another increment. It also fits into the nature of the product. We can have something on the market fast and then add functionality with each version, er, increment.
- Lee: Wait a minute, did you say that we regenerate the plan with each tour around the spiral, Doug? That's not so great, we need one plan, one schedule, and we've got to stick to it.
- Doug: That's old school thinking, Lee. Like Ed said, we've got to keep it real. I submit that it's better to tweak the plan as we learn more and as changes are requested. It's way more realistic. What's the point of a plan if it doesn't reflect reality?
- Lee (frowning): I suppose so, but senior management's not going to like this… they want a fixed plan.
- Doug (smiling): Then, you 'll have to reeducate them, buddy

KAIST