

# Rational Unified Process

## An Overview

Dave Levitt

CS 2000: Systems Analysis & Design

# Agenda

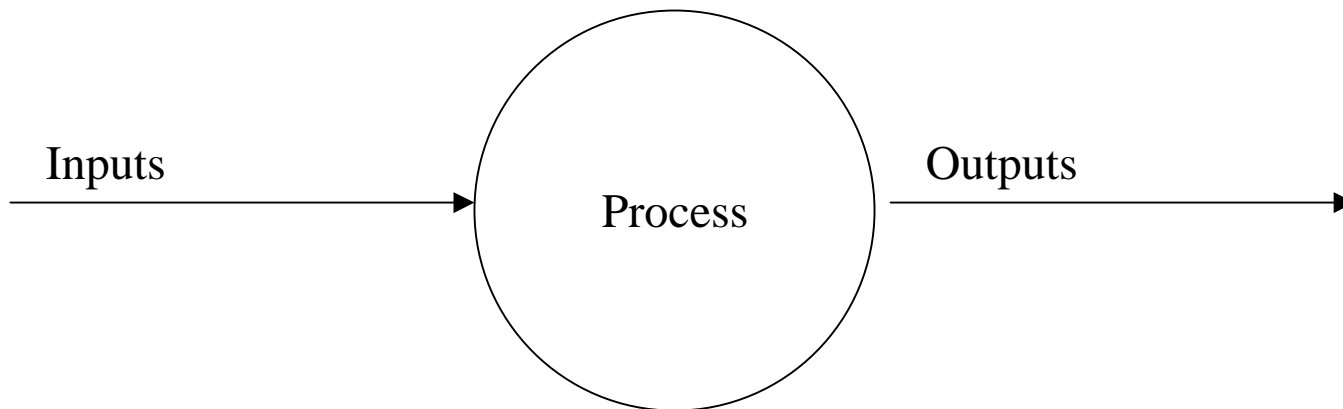
- Questions from last week?
- Discuss Articles
- Discuss Teams / Team Selection
- Discuss Software Dev. Process / RUP
- Intro. To Requirements

# Key Terms

- Requirements – capabilities that a system must fulfill.
- Analysis - a process of discovering requirements
  - OO Analysis (OOA): representing the requirements in business terms, primarily through the use of objects. Example: Customer, Invoice, Ledger
- Design – a blueprint for implementation.
  - OO Design (OOD): a blueprint represented using objects. Example: Customer, Invoice, etc.
- Unified Modeling Language (UML) – an industry standard, vendor and language neutral diagramming notation for the specification, construction, and documentation of software (OO) based systems. Managed by Object Management Group.
- Rational Rose – a software application that renders OOA & OOD Models developed in UML.

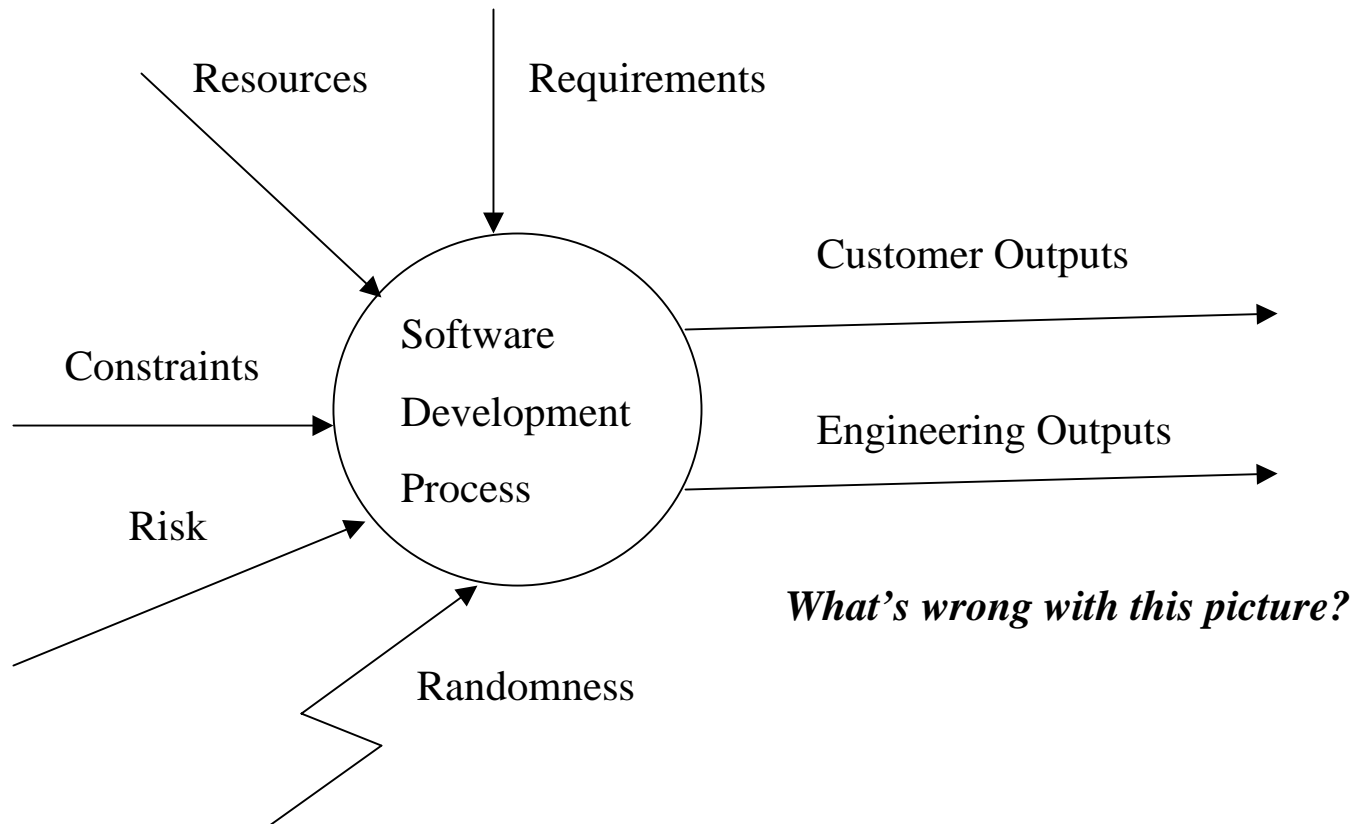
# Process Overview

- In its most abstract sense, a process is a set of automated and/or manual activities that convert inputs into outputs for the purpose of achieving a goal.



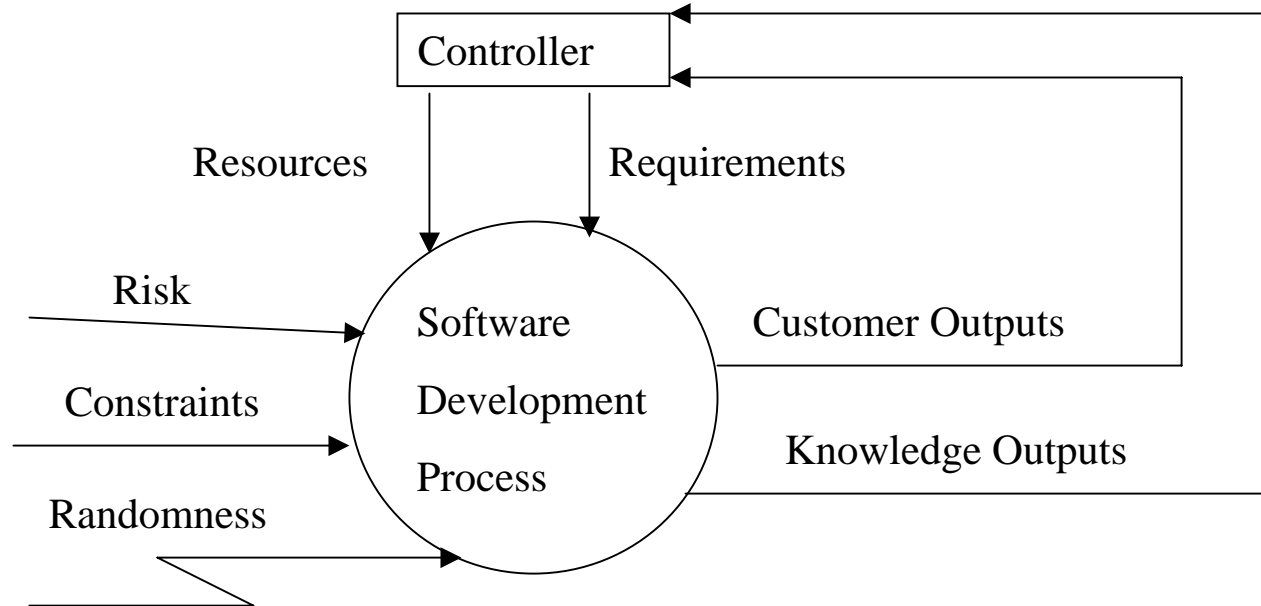
# Process Overview – cont'd

In a Software Development Process, the goal is to create or enhance a software system. The inputs include Requirements, Resources, etc. The outputs include Customer and Engineering outputs.



# Process Overview – cont'd

- The simplified process portrayed earlier will work, but it would be nice if the outputs had a feedback loop that can facilitate reuse.



# Process Overview – cont'd

- An effective process, besides being able to achieve its short-term goal(s), should also look ahead to fulfill long-term goals. With respect to a software development process...
  - What are some of the short-term goals?
  - What are some of the long-term goals?
- In its most abstract sense, a software development process is a specific implementation of any type of engineering process. The activities of any engineering process include:
  - Engineering activities:
    - Analysis, Design, Construction, Testing, Deployment, Maintenance.
  - Management activities:
    - Project Management, Configuration Management, Risk Management, etc.

# Process Overview – cont'd

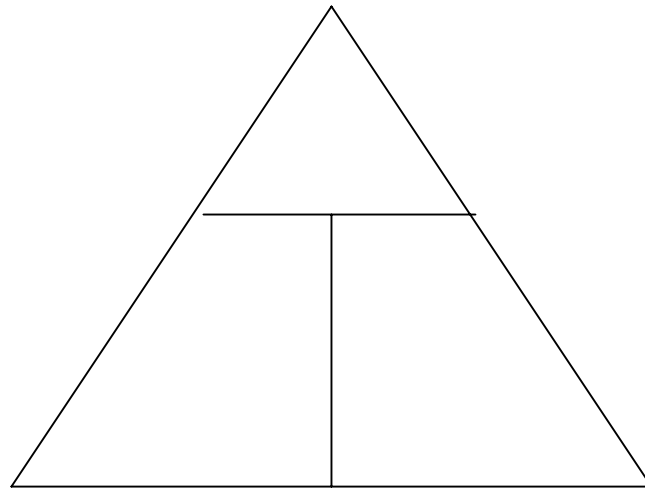
## Intermezzo Number 1

Comment on this statement:

‘The process was a success, but the project failed’

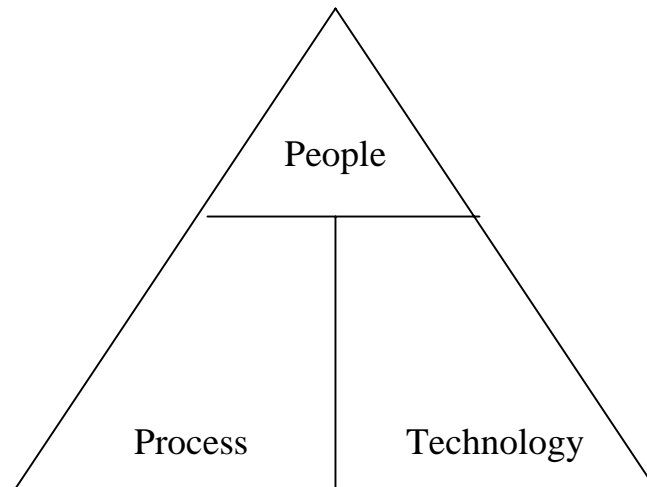
# Process Overview – cont'd

- For simplicities sake, assume there are 3 dimensions to a software project: people, process, technology. Which dimension goes into the top of the triangle?



# Process Overview – cont'd

- The 3 most important ingredients to a successful software project are **People, People, and People!**



**Why?**

# Software Development Lifecycles (SDLC)

- An SDLC is an approach/cookbook to managing software activities.
- Major approaches include:
  - Waterfall
  - Iterative
  - Hybrid
  - Other
- While the author and many other respected people advocate that only iterative projects succeed, such is not the case. Rationale:
  - People, people, people! In fact, management is the single biggest success factor!
  - Many of the worlds largest systems were developed non-iteratively!
  - Many iterative projects have failed too!

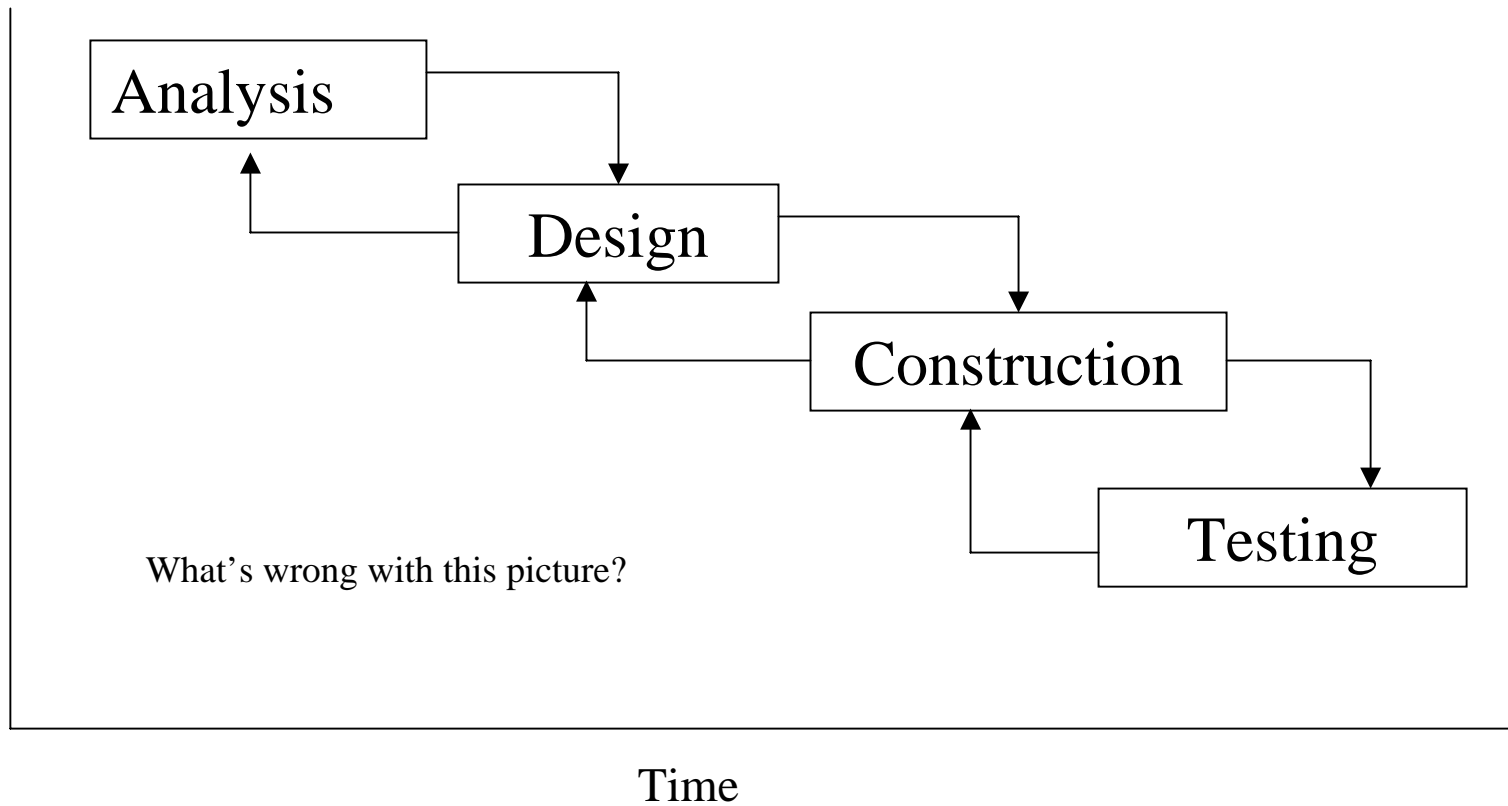
# SDLC Approaches

- Factors to consider when choosing an SDLC approach.
  - How stable are the requirements?
  - How restrictive are the constraints?
  - What are competitors doing?
  - How much risk are stakeholders willing to assume?
  - What type of project is it – conversion, new development?
  - Will I need to make midcourse corrections?
  - How much visible progress will I need for stakeholders?
  - How much experience do team members have?
- Most organizations exclusively use one and only one approach. Is this desirable?

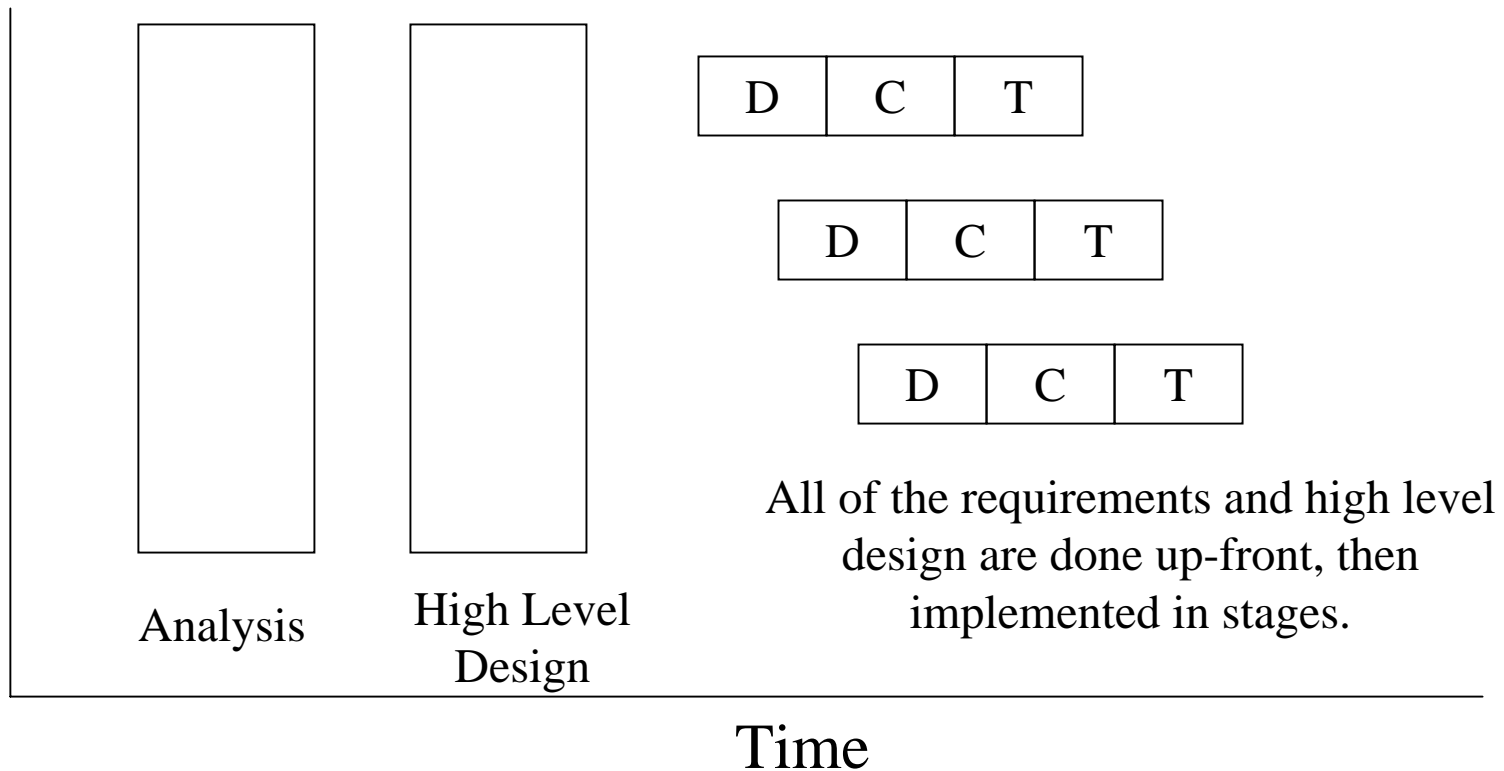
# SDLC – Waterfall

- Developed long, long ago (in a far away land).
- Based on the premise that artifacts developed for phase ‘n’ are complete for phase n+1.
- Does not work well when....
  - Requirements are changing.
  - Risk is high.
  - Stakeholders need visible progress.
  - Planning is important.
- Works well for some types of projects, like conversions.

# SDLC - Waterfall



# SDLC – Staged Delivery



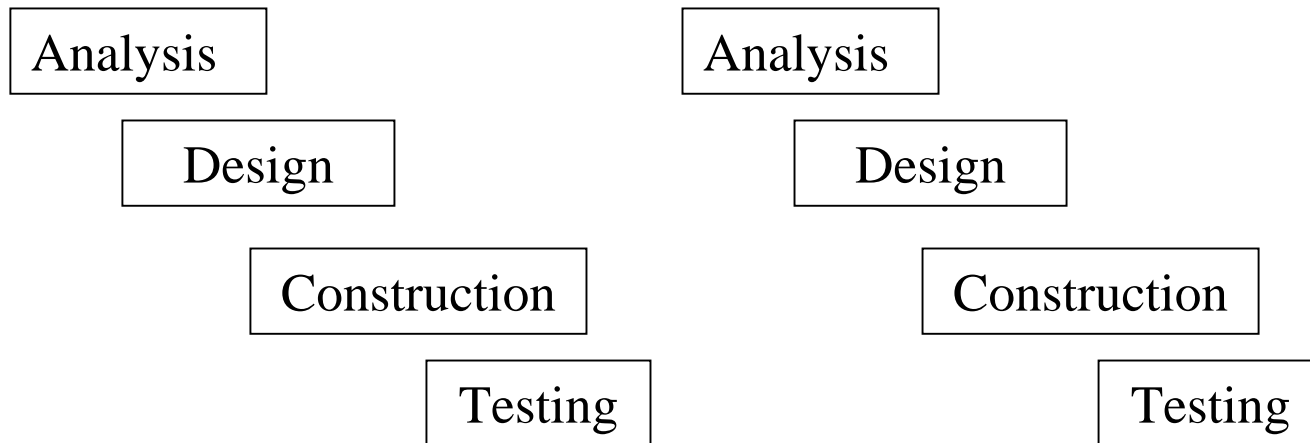
# SDLC – Staged Delivery (cont'd)

- Advantages:
  - Delivers most needed functionality first.
  - Provides a reasonable degree of visible progress.
  - Mitigates risk better than waterfall
  - Sometimes may be a viable approach if no developer resources are available right now.
- Disadvantages.
  - Requires careful planning and coordination between management and engineering teams.
  - Requirements must be reasonably stable.

# SDLC - Iterative

- Based on industry best practices: risk mitigation, visible progress, evolving requirements, time-boxing, architecture driven, feedback and adaptation.
- First introduced in late 80's by Barry Boehm. Known (Spiral Model) and Tom Gilb (Evolutionary Development)
- Other variations include UP, RUP, Agile Methods (XP, SCRUM, others).
- In this class, we focus on UP / RUP. We'll talk more about Agile Methods in CS 2100. Stay tuned!

# SDLC - Iterative (cont'd)



The activities remain the same, but are not done for the entire system, only a portion of it. An iteration is like a mini-waterfall that lasts (ideally) from 2 to 6 weeks.

Time

Weeks 2 and 3

# SDLC - Iterative (cont'd)

- Advantages:
  - Excellent for changing requirements.
  - Excellent risk mitigation.
  - Excellent visible progress.
  - Excellent for planning at the iteration level.
- Difficult for overall planning.
  - What is my total cost?
  - How long will it take until everything is done?

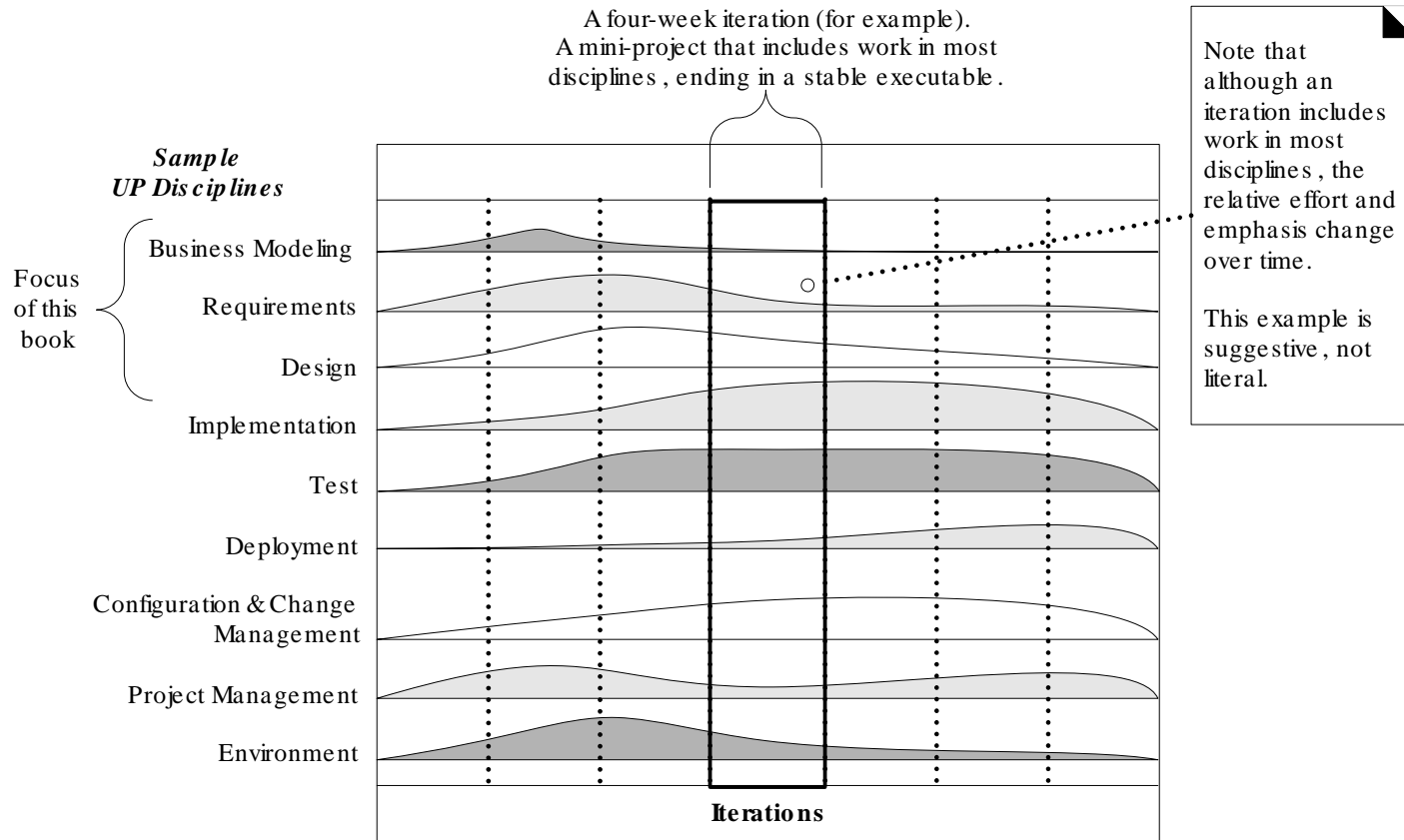
# SDLC – Other Approaches

- Commercial off the shelf (COTS)
- Code and Fix (not a strategy).

# Unified Process (UP)

- Specific implementation of iterative development.
- Based on notions of iterations, phases, and disciplines: engineering and management.
- See graphic on next page (from [www.craiglarman.com](http://www.craiglarman.com))
- The UP is a derivative of a proprietary process known as the Rational Unified Process (RUP).
  - RUP is now owned, updated, and marketed by IBM Rational.
  - Initially developed by “the 3 amigos”: Grady Booch, James Rumbaugh, Ivar Jacobson.
    - The 3 amigos also developed the initial version of the UML!

# UP (cont'd)



# UP (cont'd)

- Iterations:
  - An iteration is like a mini-waterfall.
  - The goal of each iteration is to build an incremental and working version of the system.
  - Are a slice of time.
- Iterations are subdivided into phases.
  - Inception Phase
  - Elaboration Phase
  - Construction Phase
  - Transition Phase

# UP (cont'd)

- Iteration Planning:
  - Risk drives each iteration.
  - Early iterations build the business case and address the highest (usually non-functional requirements).
  - Later iterations address specific use cases (a technique to capture functional requirements).
  - Goal is to have short iterations (2-6 weeks).
  - Not all iterations have to have the same duration.
  - What effect will team size have on iteration duration? How about team locale? Experience? Other factors?

# UP – (cont' d)

- Engineering phases include:
  - Inception
  - Elaboration
  - Construction
  - Transition
- A phase can have multiple iterations.
  - Influenced by risk, project size, project constraints, etc.
  - Typically the Inception phase has only 1 iteration.

# UP (cont'd)

- The goal of the Inception phase is to determine if the project should proceed.
  - Establish the vision / business case.
  - Establish scope (use case model, maybe a domain model).
  - Develop initial set of use cases and related artifacts.
  - Provide rough estimates.
- The goal of the Elaboration Phase is to build the architecture that supports later iterations.
  - Refine Inception Phase activities.
  - Resolve high risk non-functional and functional requirements.
  - Re-estimate based on better understanding of the requirements

# UP (cont'd)

- The goal of the Construction phase is to implement (program) the lower risk and/or remaining use cases.
- The goal of the Transition phase is to put the iteration into a production environment.
  - Important Note: While the goal of each phase is to produce an executable release for testing and customer feedback, interim releases from earlier phases are not placed in the production environment. They are typically placed in the Test environment, or something like that.

# UP (cont'd)

- Looking at the humpback whale chart:
  - What are the primary disciplines in the Inception Phase?
    - How about the other phases?
  - When are most of the requirements captured?
  - What management process would deal with changing requirements?
  - Where does most of the OOA/D activities take place?

# UP (cont'd)

- Looking at the humpback whale chart (cont'd)
  - Which artifacts would you typically produce in the Inception Phase? The Elaboration Phase?
  - Which UP document specifies which artifacts are required and optional? Which management discipline might this be done in?

# Concluding Thoughts

- Software processes are a hot topic, with many books, articles, conferences, and speaking engagements. Each seeks to document that “their” process is the best (or at least better than anybody else’s).
- Just like there is no ideal programming language, there is no ideal process. Any claims to the contrary is snake oil.
- Process, despite its importance, is only one dimension to a successful software project, yet it baffles me that the the most important dimension, which is ???????, does not receive the same amount attention.
- I can only conclude that process is easy (and perhaps lucrative), but dealing with the human dimension is not!

# Concluding Thoughts (cont'd)

- Comment on this statement:

“People are our most important asset”