CSCI222 System Development

Software process models

This lecture …

- Software process models
  - Waterfall model
  - Prototyping model
  - Iterative/incremental model
  - Spiral model
  - Rational Unified Process
  - Agile methods

Acknowledgement: some materials are adapted from Chapter 2 - Ian Sommerville (2010), *Software Engineering,* 9th Edition, Addison-Wesley.

The software process

- A **structured set of activities** required to develop a software system

- Many different processes but they all involve:
  - Requirements specification- defining what the system should do;
  - Design & Implementation - defining the organization of the system and implementing the system
  - Verification & Validation (V & V) - checking that it conforms to the specification and does what the customer wants
  - Maintenance and Evolution - changing the system in response to changing customer needs.

- A software process model is an **abstract representation of a process**. It presents a description of a process from some particular perspective

Waterfall model

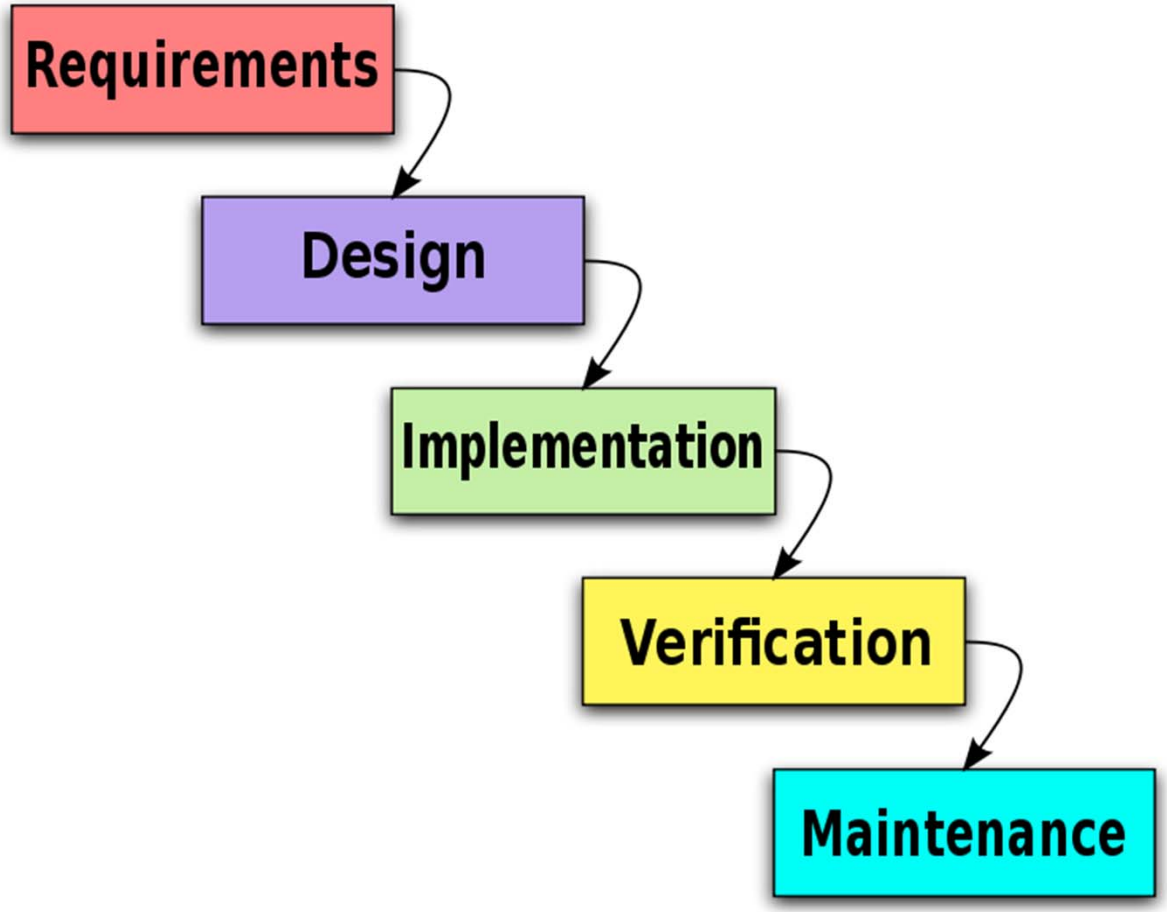
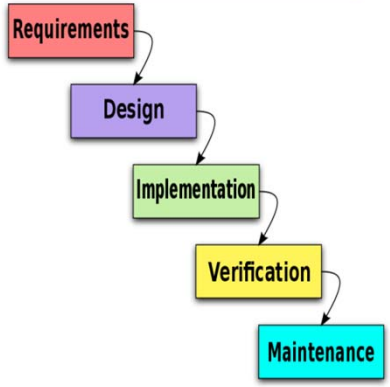


Image source: http://en.wikipedia.org/wiki/Waterfall_model

Waterfall model (cont.)

- The waterfall model is a sequential software development process.
  - a phase has to be *complete* and absolutely *correct* before moving onto the next phase.
- Progress flows from the top to the bottom, like a waterfall.
- Has its origins in the manufacturing and construction industries and adapted for software development since 1970s.
- First described for software development by **Winston W. Royce** in 1970



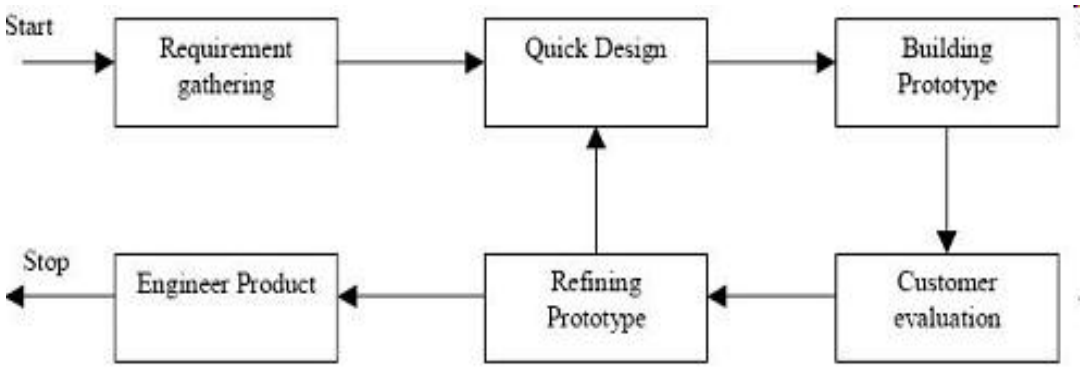
Waterfall model limitations

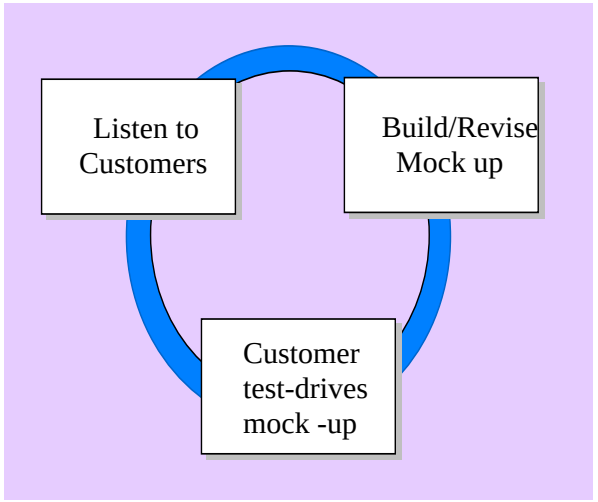
- The difficulty of responding to changing customer requirements.
- Designers will have to fully predict problem areas of the system and produce a *correct* design before implementation is started
- This model is only appropriate when the requirements are well-understood and changes will be fairly limited during the design process.
  - Few business systems have stable requirements.

YouTube
The Rise and Fall of Waterfall

http://www.youtube.com/watch?v=X1c2--sP3o0

Prototyping model





- gather requirements
- quick design focusing on what will be visible to user – input & output formats
- build a prototype, i.e. a working version of the system
- prototype evaluated and requirements refined

- process iterated until customer & developer satisfied
  - then throw away prototype and rebuild system to high quality

Throw-away prototypes

- Prototypes should be discarded after development as they are not a good basis for a production system:
  - It may be impossible to tune the system to meet non-functional requirements;
  - Prototypes are normally undocumented;
  - The prototype structure is usually degraded through rapid change;

Prototyping model
Benefits

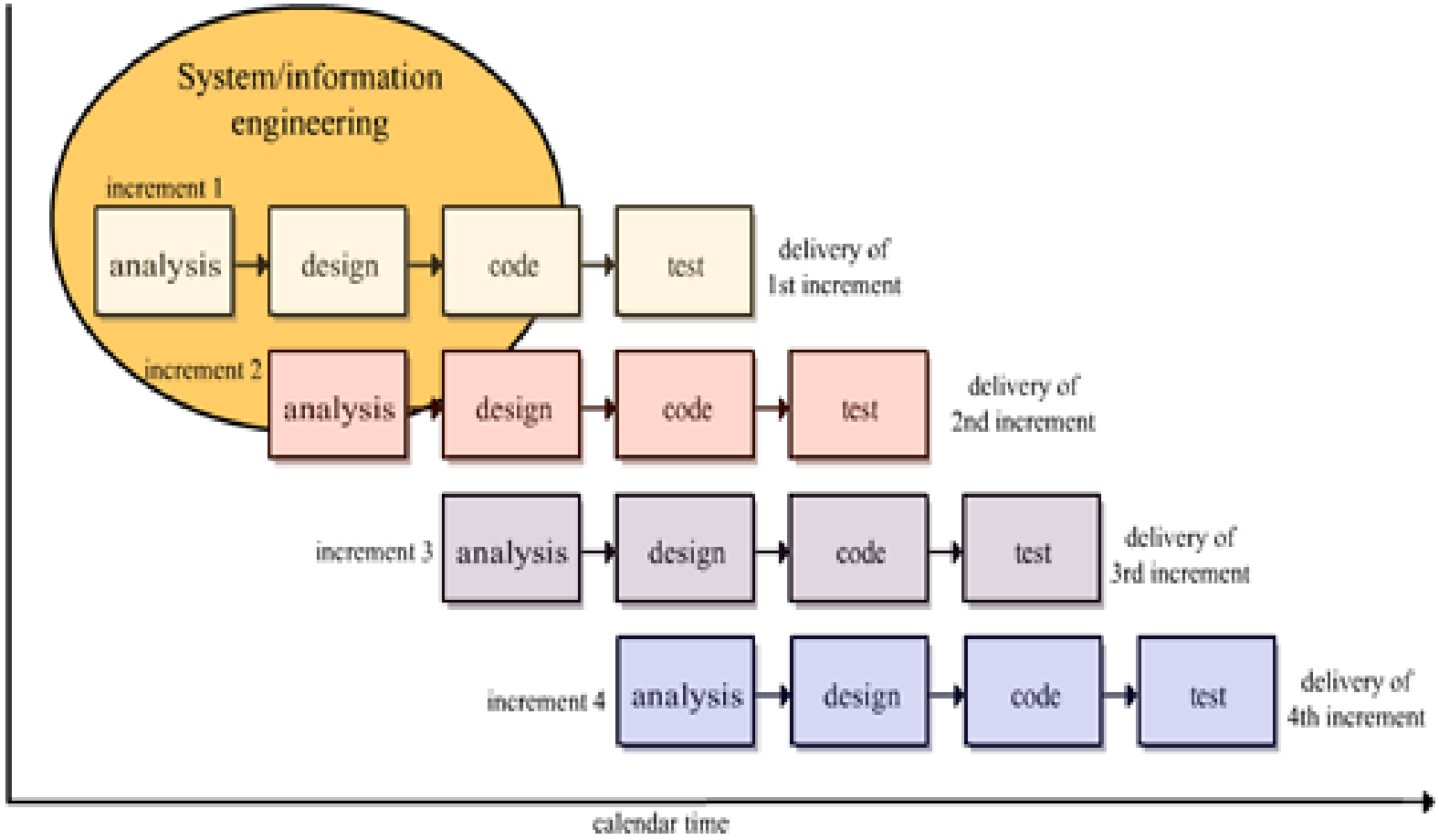
- Users are actively involved in the development
- Reduced time and costs
  - a working model of the system is provided, the users get a better understanding of the system being developed.
  - Prototyping can improve the quality of requirements and specifications provided to developers.
  - Quicker user feedback is available leading to better solutions.
- Best projects to use prototyping
  - Human-Computer Interface

Prototyping model Limitations

- User confusion of prototype and completed system
  - Users can begin to think that a prototype, intended to be thrown away, is actually a final system that merely needs to be finished or polished.
- Developer attachment to prototype:
  - Developers can also become attached to prototypes they have spent a great deal of effort producing; this can lead to problems like attempting to convert a limited prototype into a final system when it does not have an appropriate underlying architecture
- Excessive development time of the prototype:
  - A key property to prototyping is the fact that it is supposed to be done quickly. If the developers lose sight of this fact, they very well may try to develop a prototype that is too complex
- Expense of implementing prototyping:
  - the start up costs for building a development team focused on prototyping may be high

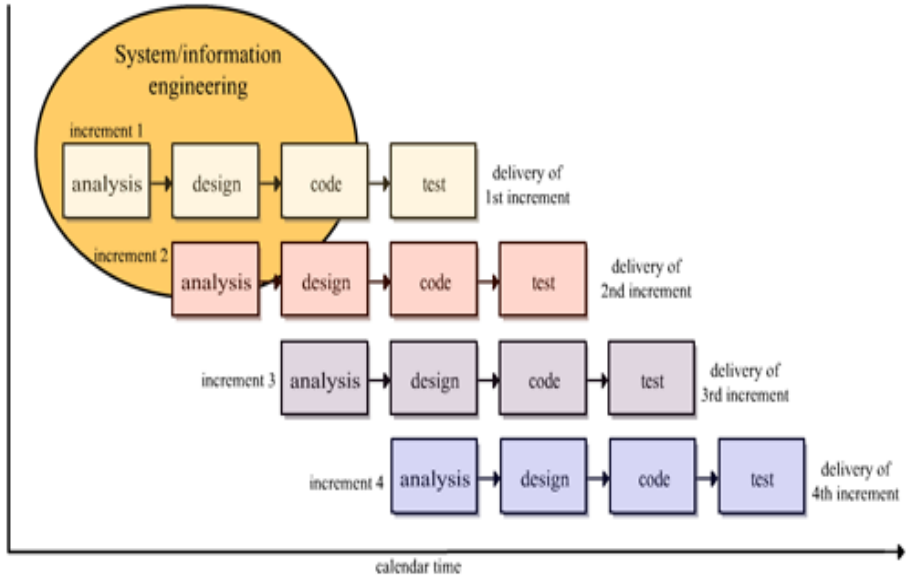
Source: http://en.wikipedia.org/wiki/Software_prototyping

Incremental and Iterative model

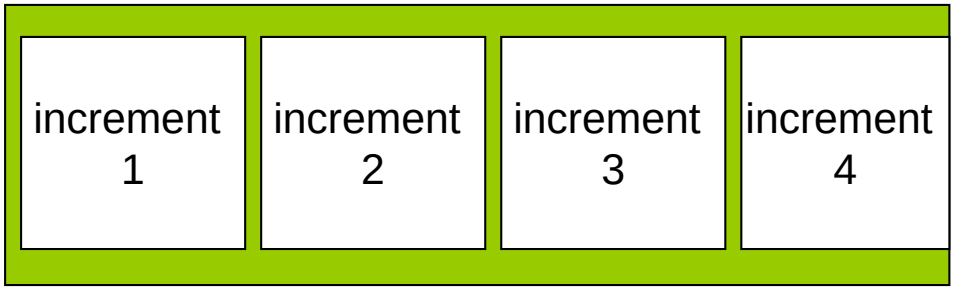


Incremental and Iterative model

- Develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).
- Rather than deliver the system as a single delivery, the development and delivery is broken down into increments with each increment delivering part of the required functionality.
- User requirements are prioritised and the highest priority requirements are included in early increments.
- At each iteration, design modifications are made and new functional capabilities are added.



The software system



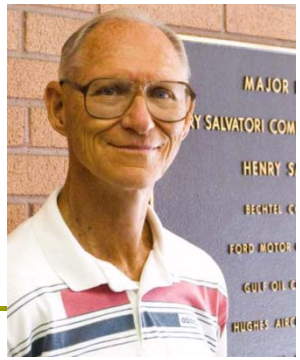
Incremental approach:benefits

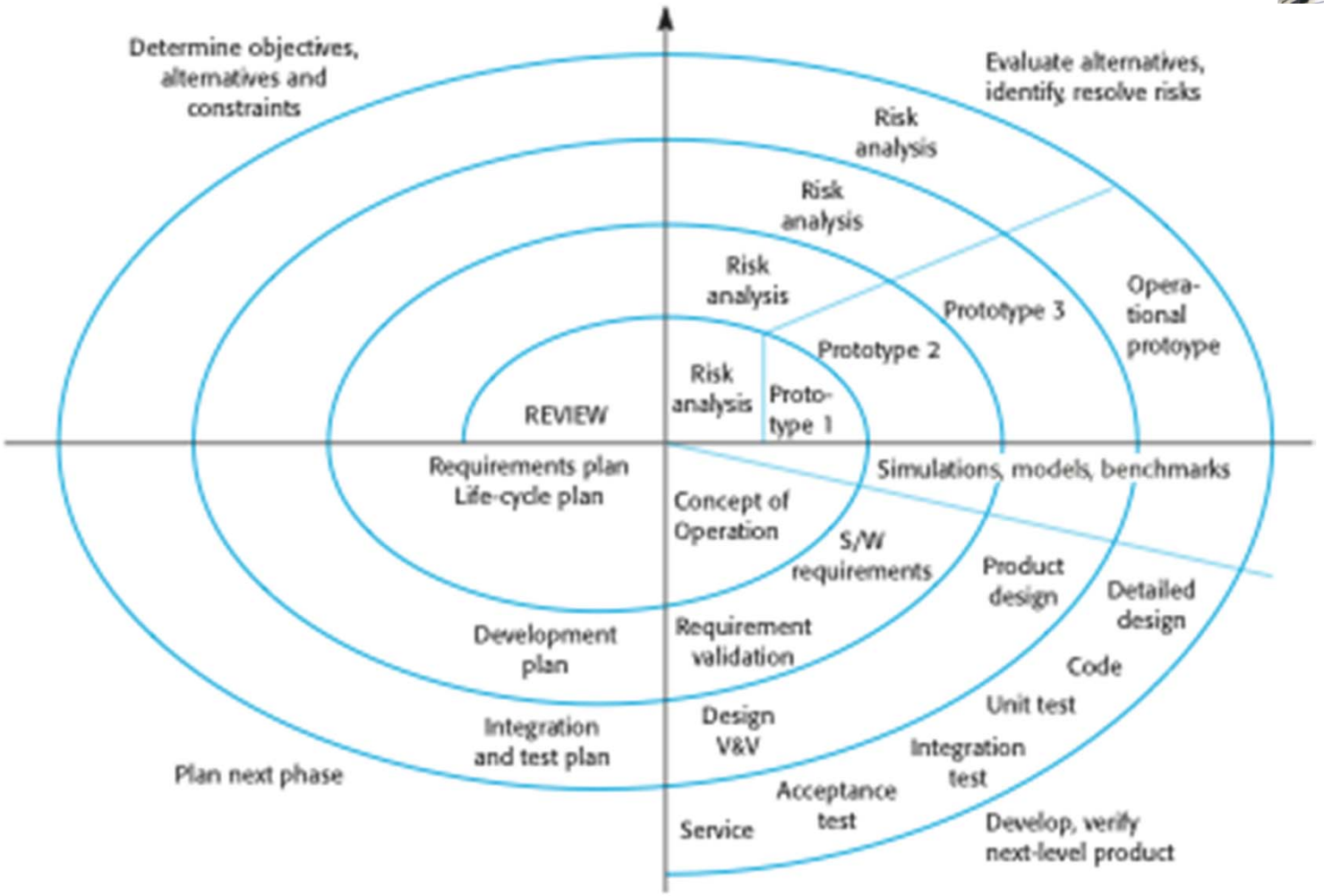
- Feedback from early stages used in developing latter stages
- The cost of accommodating changing customer requirements is reduced.
- More rapid delivery and deployment of useful software to the customer is possible.
  - Customers are able to use and gain value from the software earlier than is possible with a waterfall process
- Early increments act as a prototype to help elicit requirements for later increments.
- Lower risk of overall project failure.
- The highest priority system services tend to receive the most testing.

Possible disadvantages of incremental delivery

- Loss of economy of scale
  - some costs will be repeated
- System structure tends to degrade as new increments are added.
  - Unless time and money is spent on refactoring to improve the software, regular change tends to corrupt its structure.
  - Incorporating further software changes becomes increasingly difficult and costly.

The Spiral Model





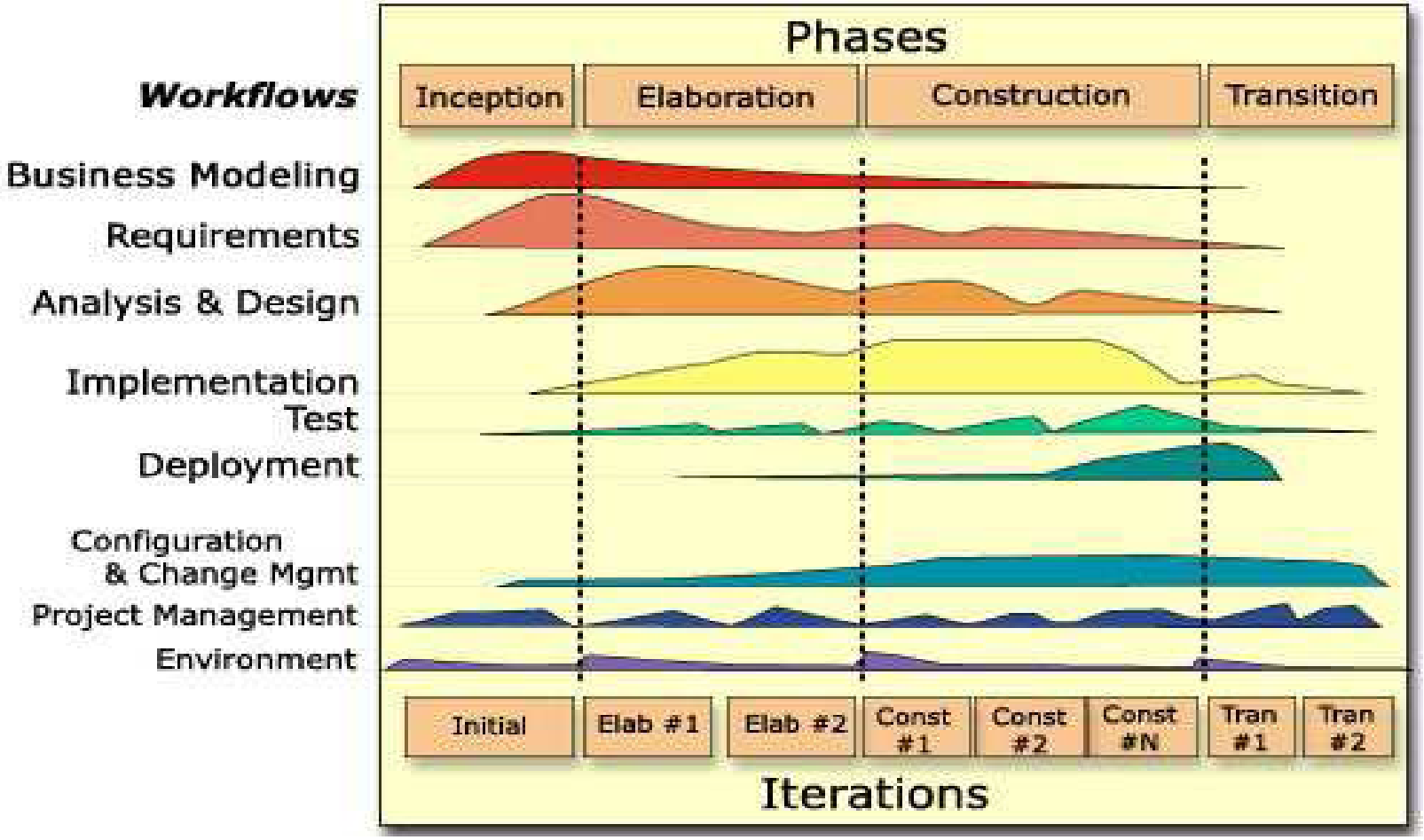
The Spiral Model

- A combination of prototyping with iterative model
- Process is represented as a spiral rather than as a sequence of activities with backtracking.
- Each loop in the spiral represents a phase in the process.
- Risks are explicitly assessed and resolved throughout the process.

Spiral model usage

- Spiral model has been very influential in helping people think about iteration in software processes and introducing the risk-driven approach to development.
- The spiral model is mostly used in large projects.
  - Game development is a main area where the spiral model is used and needed
  - Military projects, e.g. Future Combat Systems program.
    - it had a 2 year iteration (spiral).

The Rational Unified Process



Agile methods

- Dissatisfaction with the overheads involved in software design methods of the 1980s and 1990s led to the creation of agile methods. These methods:
  - **Focus on the code** rather than the design
  - Are based on an **iterative** approach to software development
  - Are intended to **deliver working software** quickly and evolve this quickly to meet changing requirements.

Agile methods and Scrum

- Watch the following video and answer the questions in the next slide

http://www.youtube.com/watch?v=OJfIDE6OaSc

Agile methods
Scrum

1) Which one of the following is NOT a Scrum role?

A. Product Owner
B. ScrumMaster
C. Product Manager
D. Team

2) Which of the following meeting is NOT part of Scrum?

A. Product review meeting
B. Sprint review meeting
C. Sprint planning meeting
D. Sprint retrospective meeting

3) Which of the following questions is not relevant for regular Scrum Meeting?

A. What have you done since the last Daily Scrum regarding this project?
B. What will you do between now and the next Daily Scrum meeting regarding this project?
C. Would you be able to finish your work on time?
D. What impedes you from performing your work as effectively as possible?

4) What is a Burndown Graph?

A. A Sprint plan which is burned to celebrate successful completion of Sprint
B. The trend of work remaining across time in a Sprint, a release, or a product
C. The trend of work accomplished across time in a Sprint, a release, or a product
D. A Graph to measure human Burnout effect due to fast pace of sprint

5) Name the meeting during which the Team demonstrates to the Product Owner and any other interested parties what it was able to accomplish during the Sprint.

A. Sprint retrospective meeting
B. Product review meeting
C. End-of-Sprint review meeting
D. Stakeholder Review Meeting