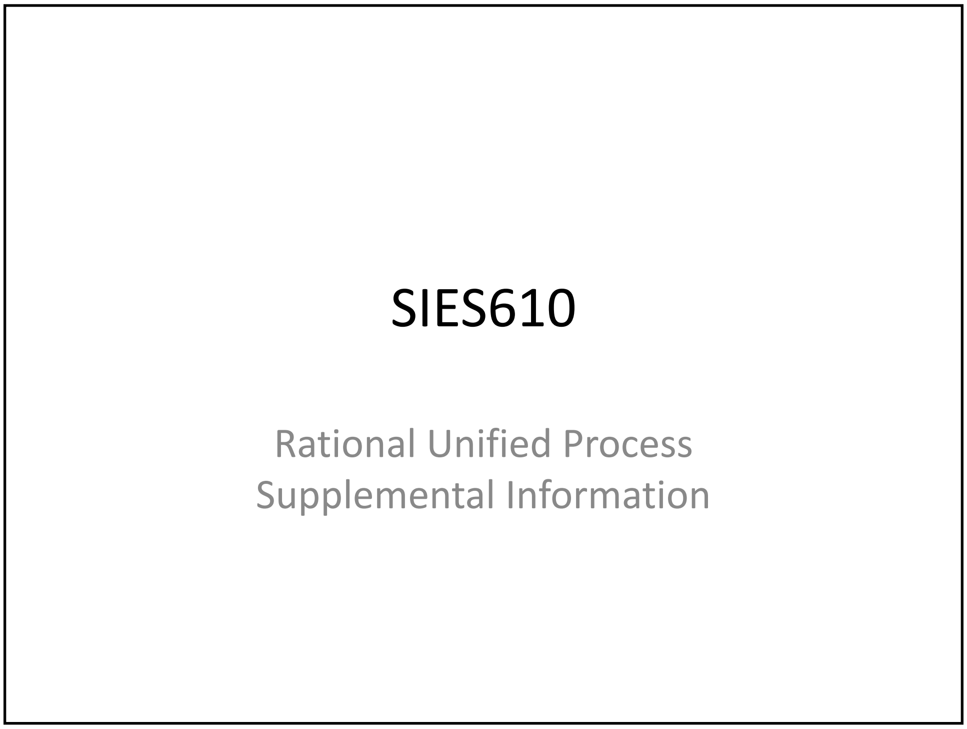
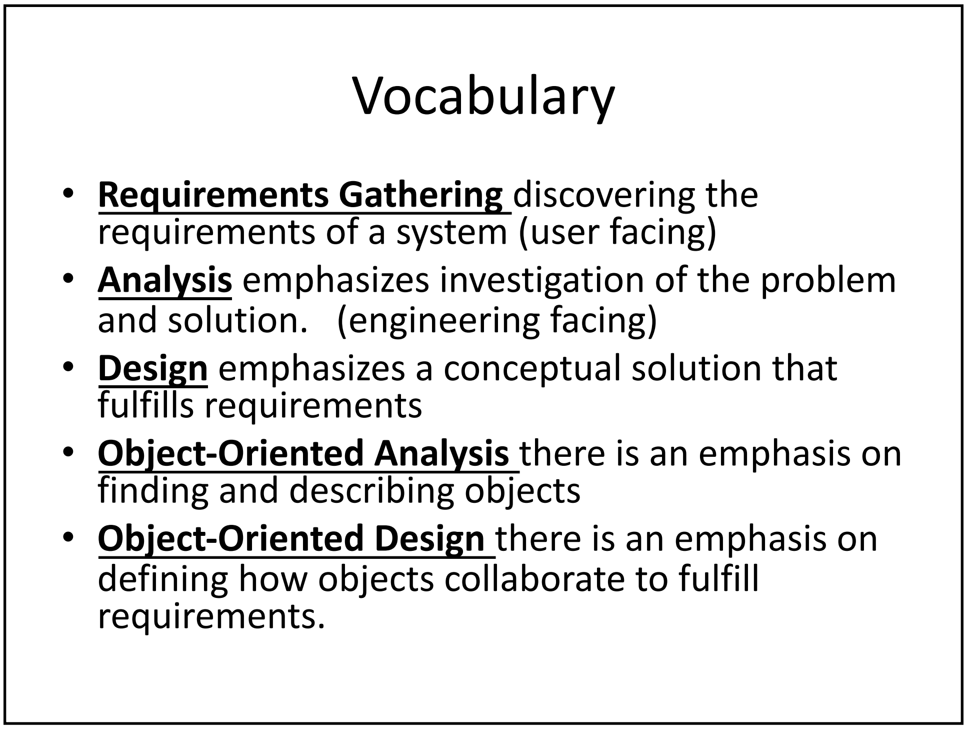


SIES610

Rational Unified Process
Supplemental Information

1



Vocabulary

- **Requirements Gathering** discovering the requirements of a system (user facing)
- **Analysis** emphasizes investigation of the problem and solution. (engineering facing)
- **Design** emphasizes a conceptual solution that fulfills requirements
- **Object-Oriented Analysis** there is an emphasis on finding and describing objects
- **Object-Oriented Design** there is an emphasis on defining how objects collaborate to fulfill requirements.

2

More Vocabulary

- Functional Requirement
  - Something the product must do.
- Non-Functional Requirement
  - Something the product will be judged by.
  - Example: The product must be secure.
  - Example: The product must be easy to use.
  - Example: The product will be used in libraries so should not have sound.

3

Agile

- Agile applies to time-boxed, iterative (i.e. *iterations*) and evolutionary development.
- There are several approaches to agile
  - Scrum
  - Extreme programming
  - Etc.
- Unified process is only one of them
- Agile modeling likes UML as a sketch

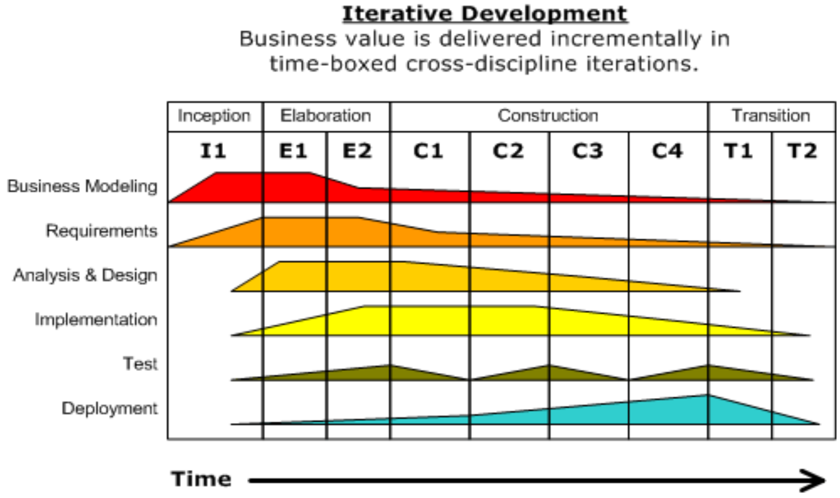
4

Unified Process Phases (UP)

- **Inception** – Vision, business case, requirements, limitations/boundaries, vague estimates
- **Elaboration** – refine vision, emphasis on analysis and design, other workflows can happen
- **Construction** – Iterative implementation, but other workflows still happen
- **Transition** – Passing to customer, emphasis is on the handoff.

5

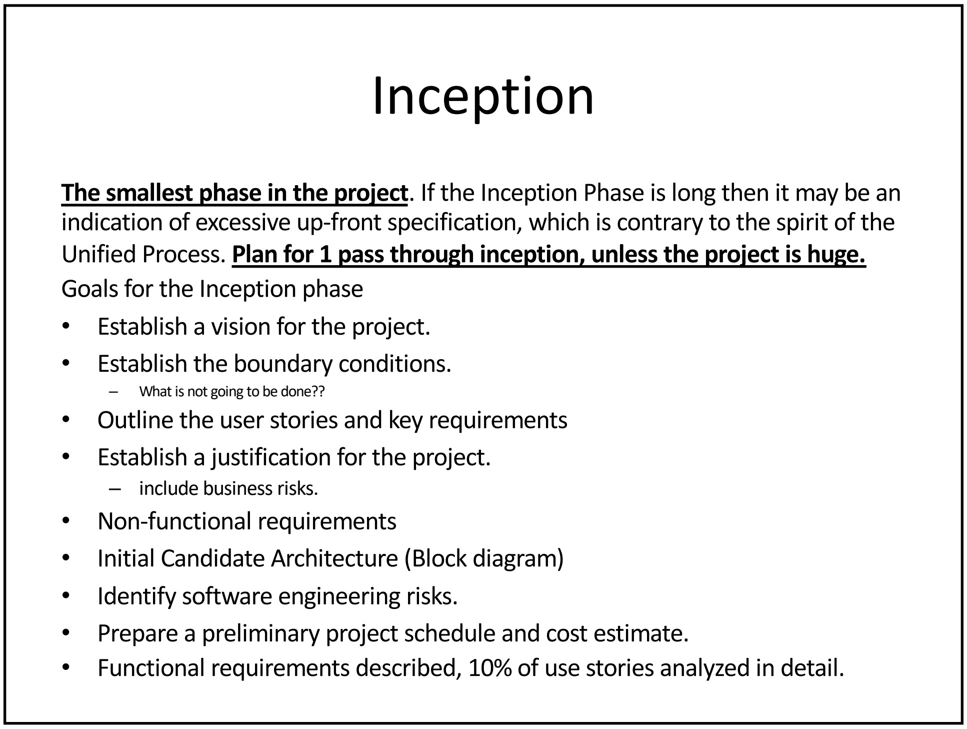
Iterative Development



Iterative Development

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

6



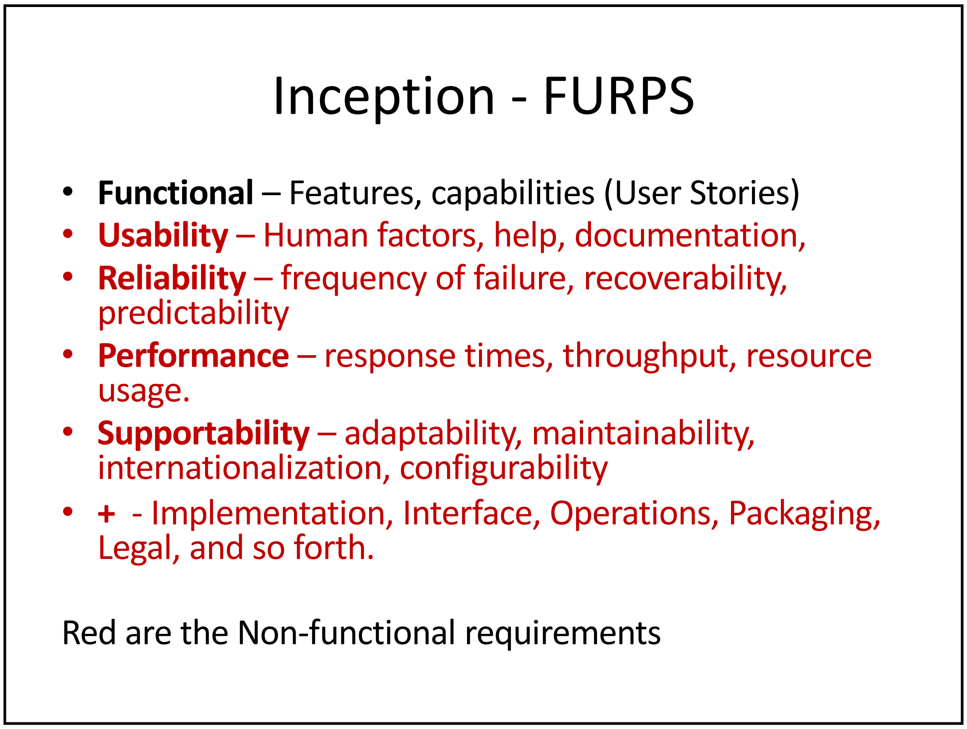
Inception

<u>The smallest phase in the project</u>. If the Inception Phase is long then it may be an indication of excessive up-front specification, which is contrary to the spirit of the Unified Process. **<u>Plan for 1 pass through inception, unless the project is huge.</u>**

Goals for the Inception phase

- Establish a vision for the project.
- Establish the boundary conditions.
  - What is not going to be done??
- Outline the user stories and key requirements
- Establish a justification for the project.
  - include business risks.
- Non-functional requirements
- Initial Candidate Architecture (Block diagram)
- Identify software engineering risks.
- Prepare a preliminary project schedule and cost estimate.
- Functional requirements described, 10% of use stories analyzed in detail.

7



Inception - FURPS

- **Functional** – Features, capabilities (User Stories)
- **Usability** – Human factors, help, documentation,
- **Reliability** – frequency of failure, recoverability, predictability
- **Performance** – response times, throughput, resource usage.
- **Supportability** – adaptability, maintainability, internationalization, configurability
- **+** - Implementation, Interface, Operations, Packaging, Legal, and so forth.

Red are the Non-functional requirements

8

Inception

- Use cases/User Stories
  - Lots of use case names/features identified
  - Maybe analyze 10% of them
- Non-functional requirements
- Maybe: Rough draft of user interface
  - Hand drawn pictures
- Business Case
  - Wild estimate of cost

9

Sprints

- ‘Time boxed’ iterations
  - 2 to 6 weeks, but keep closer to 2
  - Keep consistent. If your time box is 2 weeks, keep it 2 weeks.

10

Elaboration

- A healthy majority of the system requirements should be captured.
- Requirements are analyzed
- known risk factors are addressed and the system architecture is validated
- Common processes undertaken in this phase include the creation of architectural diagrams.
- Spend more time on analysis and design and less on requirements
- Perhaps some implementation
- End of the Elaboration phase
  - the system architecture is stabilized
  - If prototypes exist, they demonstrate that the architecture will support the key functionality **<u>and mitigate risk factors.</u>**

11

Elaboration

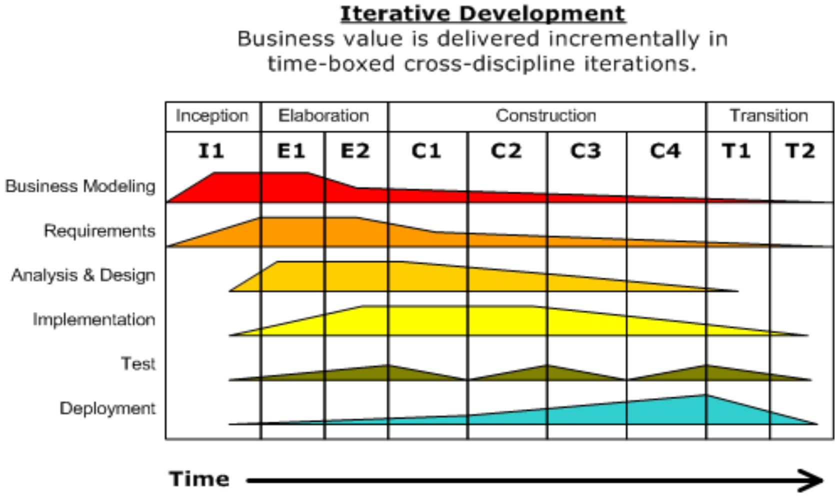
- Re-Read user stories
- Write a play (analysis)
- Draw sequence diagrams (design)
- Turn sequence diagrams into class diagrams (design)
- Repeat

12

Construction Phase

- Construction should be the largest phase in the project. In this phase the remainder of the system is built on the foundation laid in Elaboration.
- System features are implemented in a series of short, timeboxed iterations. (Sprints)
- Each iteration results in an executable release of the software.
- Our UML (Unified Modelling Language) diagrams used during this will be put to good use here

Iterative Development



Iterative Development

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

Transition Phase

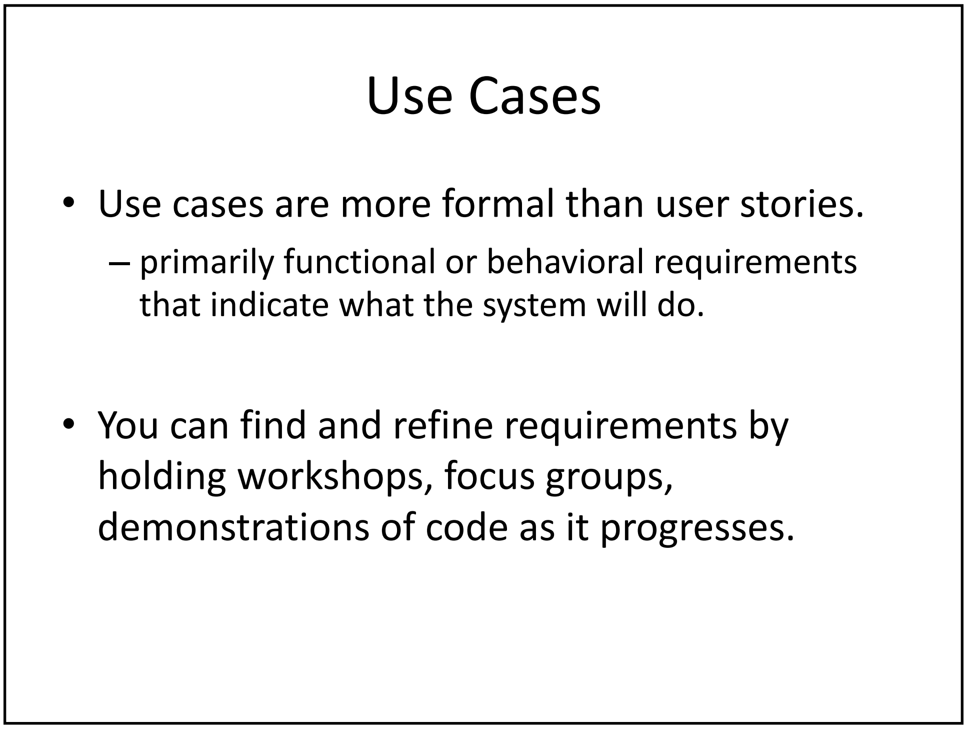
- The final project phase.
- In this phase the system is deployed to the target users.
- Feedback received may result in further refinements
- Transition phase is also built on iterations.
- The Transition phase also includes system conversions and user training.

15

User Stories

- User Stories form the functional requirements
- Michael Cohn User Story Template:
  - From here:
  - https://www.mountaingoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template
- “As a <type of user>, I want <some goal> so that <some reason>.”
  - [The ‘so that’ is **NOT** optional.]
- For example:
- As a **moderator** I want to create a new game by entering a game and an optional description so that I can start inviting estimators.

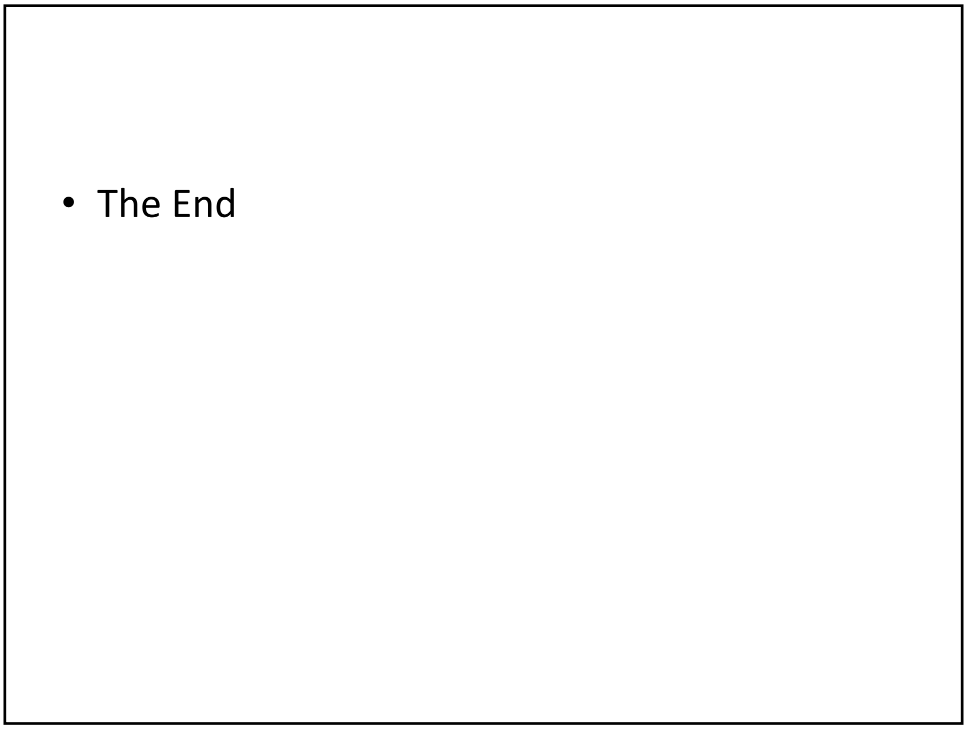
16



Use Cases

- Use cases are more formal than user stories.
  - primarily functional or behavioral requirements that indicate what the system will do.
- You can find and refine requirements by holding workshops, focus groups, demonstrations of code as it progresses.

17



- The End

18