ADVANCES IN DOMAIN

Our syllabus: RUP BY UML ?

Advances in Domain

1.AUP Scott W. Ambler ?

2.Web Engineering,

3.Reengineering

4.DevOps life Cycle

5.Towards Open Applications

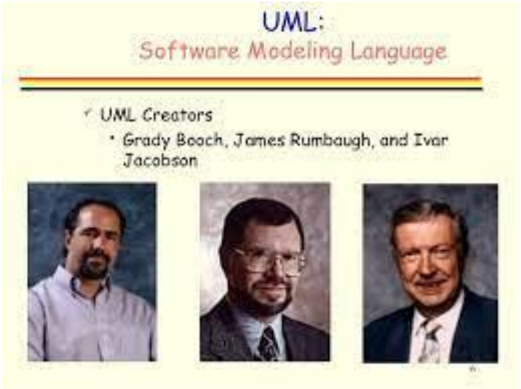
Dr. Bandu B. Meshram,
PhD(Computer Engineering),
LLM (Constitutional Law), CHFI
Professor and Former Head
Department of Computer Technology,
VJTI, Matunga,Mumbai-19
bbmeshram@ce.vjti,ac.in (9022358788)

Keep in Touch

ADVANCES IN DOMAIN

RUP BY UML ?

AUP Scott W. Ambler ?

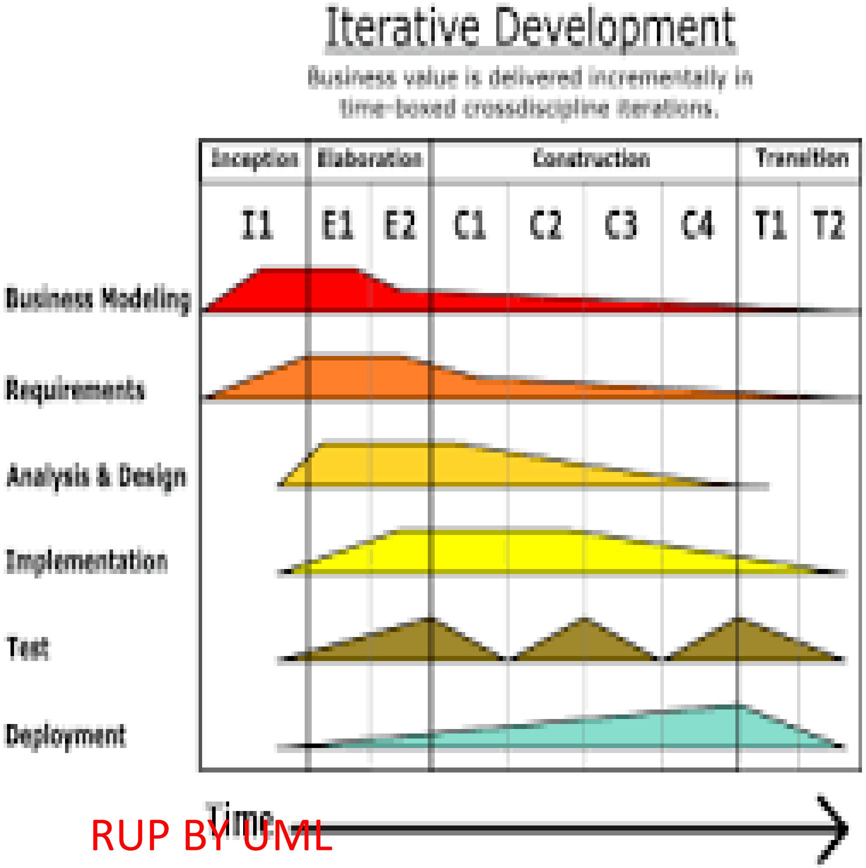


UML:
Software Modeling Language

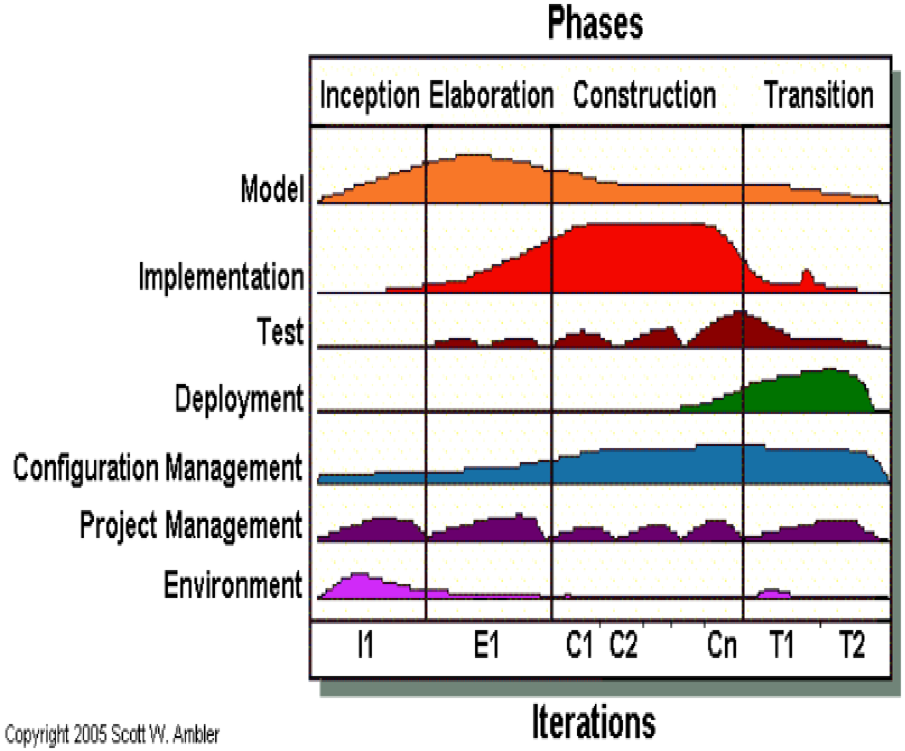
- UML Creators
  - Grady Booch, James Rumbaugh, and Ivar Jacobson

RUP BY UML

AUP BY SCOTT W. AMBLER

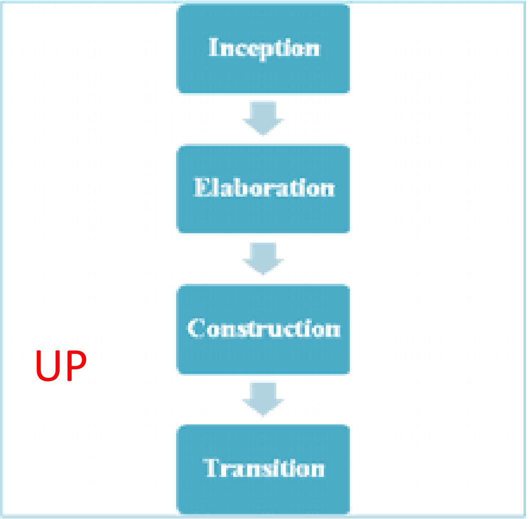


RUP BY UML

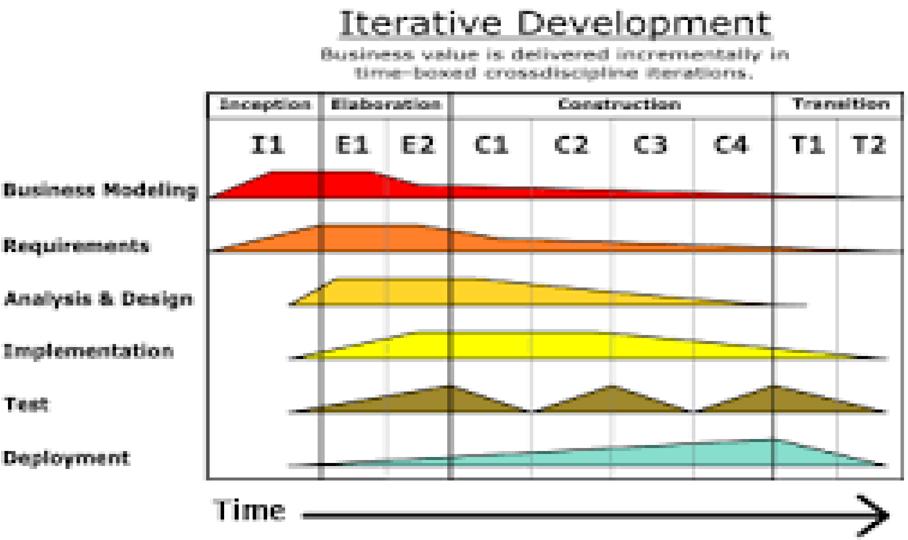


Copyright 2005 Scott W. Ambler

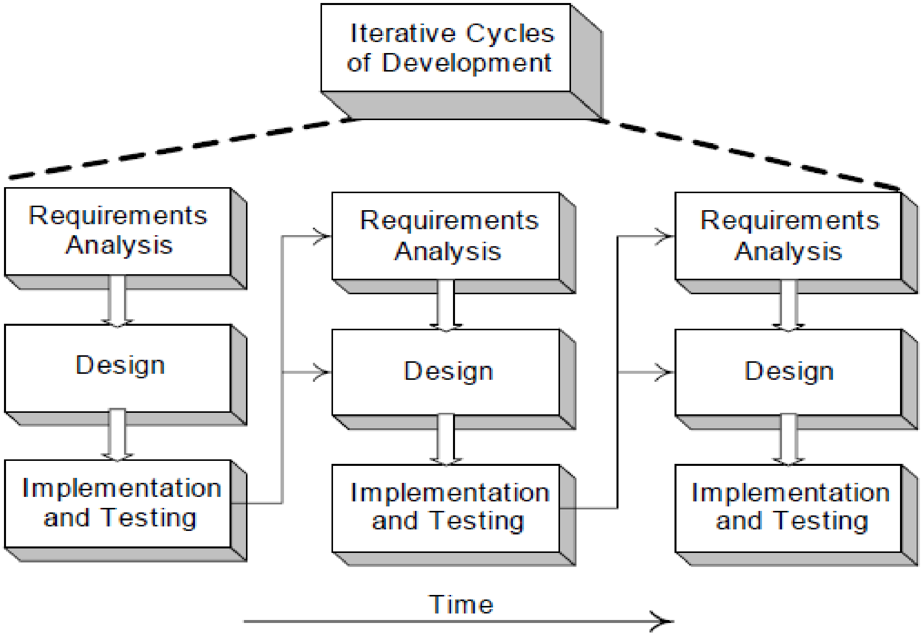
AUP BY SCOTT W.AMBLER

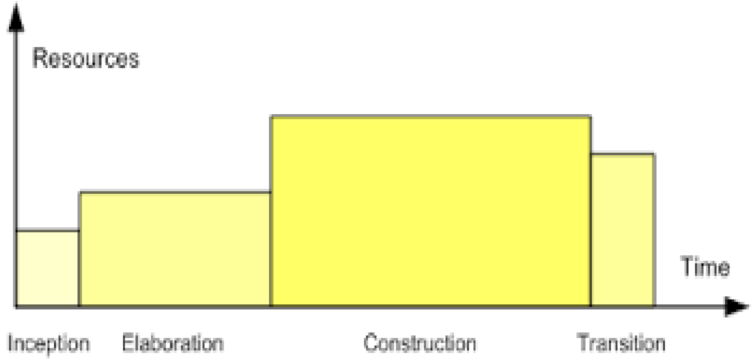

Inception
Elaboration
Construction
Transition

UP


Iterative Development
Business value is delivered incrementally in time-boxed crossdiscipline iterations.
Inception
Elaboration
Construction
Transition
I1
E1
E2
C1
C2
C3
C4
T1
T2
Business Modeling
Requirements
Analysis & Design
Implementation
Test
Deployment
Time

RUP


Iterative Cycles of Development
Requirements Analysis
Design
Implementation and Testing
Requirements Analysis
Design
Implementation and Testing
Requirements Analysis
Design
Implementation and Testing
Time


Resources
Time
Inception
Elaboration
Construction
Transition

RUP MODEL

https://www.youtube.com/watch?v=M6Mw_wDJJp0

- A UP project organizes the work and iterations across four major phases:

1. **Inception**— approximate vision, business case, scope, vague estimates AND Feasibility study –Technical, economical, operational, legal and schedule and rough estimate-N, E, D

Buy and/or build?, Should we proceed or stop?

2. Elaboration—refined vision, iterative implementation of the core architecture, resolution of high risks (high risk issues are mitigated.), identification of most requirements and scope, more realistic estimates.

3. Construction—iterative implementation of the remaining lower risk and easier elements, and preparation for deployment.

4. Transition—beta tests, deployment.

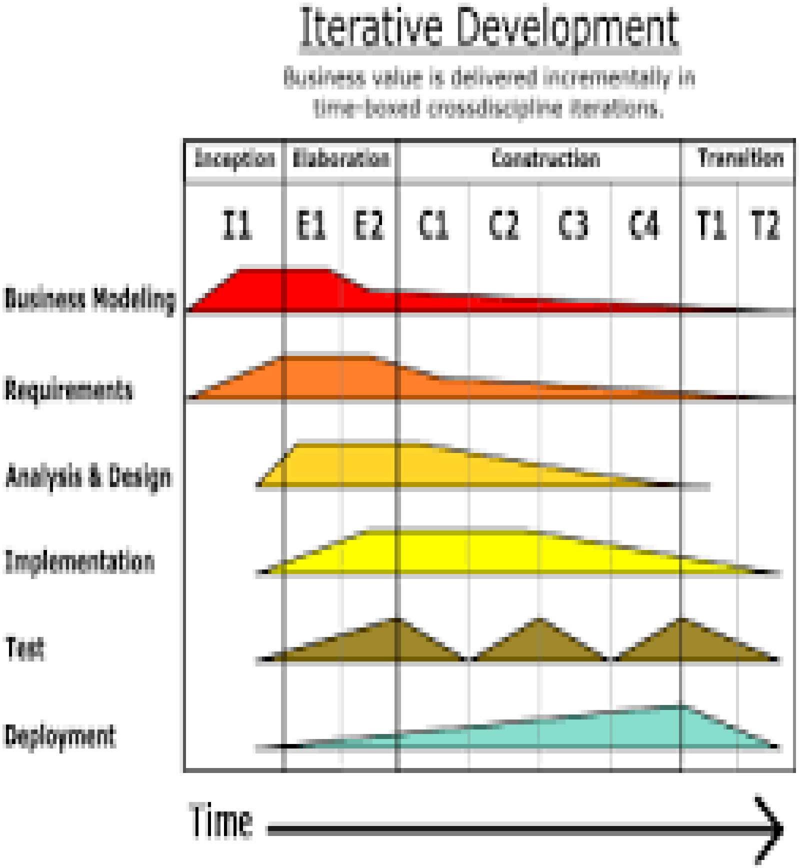
Inception is not a requirements phase; rather, it is a kind of feasibility phase, where just enough investigation is done to support a decision to continue or stop.

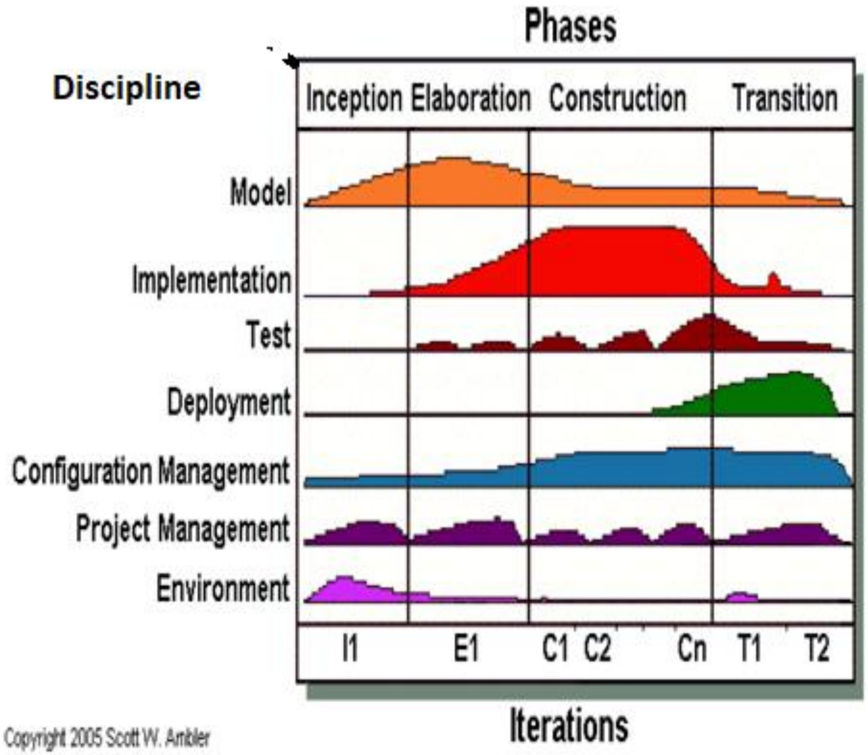
Similarly, elaboration is not the requirements or design phase; rather, it is a phase where the core architecture is iteratively implemented, and high risk issues are mitigated.

INCEPTION :Feasibility Study

- Feasibility Study is less considered feasibility analysis.
- **Technical Feasibility –**
  both hardware software
- technical skills and capabilities of technical team, existing technology can be used or not, maintenance and up-gradation is easy or not
- **Operational Feasibility –**
  In Operational Feasibility degree of providing service to requirements
- operate and maintenance after deployment.
- ,Determining suggested solution by software development team is acceptable or not etc.
- **Economic Feasibility –**In Economic Feasibility study cost and benefit of the project is analyzed.
- **Legal Feasibility –**
  This includes analyzing barriers of legal implementation of project, data protection acts or social media laws, project certificate, license, copyright etc. **Schedule Feasibility –**
  In Schedule Feasibility Study mainly timelines/deadlines is analyzed.

AUP BY SCOTT W.AMBLER





RUP BY UML

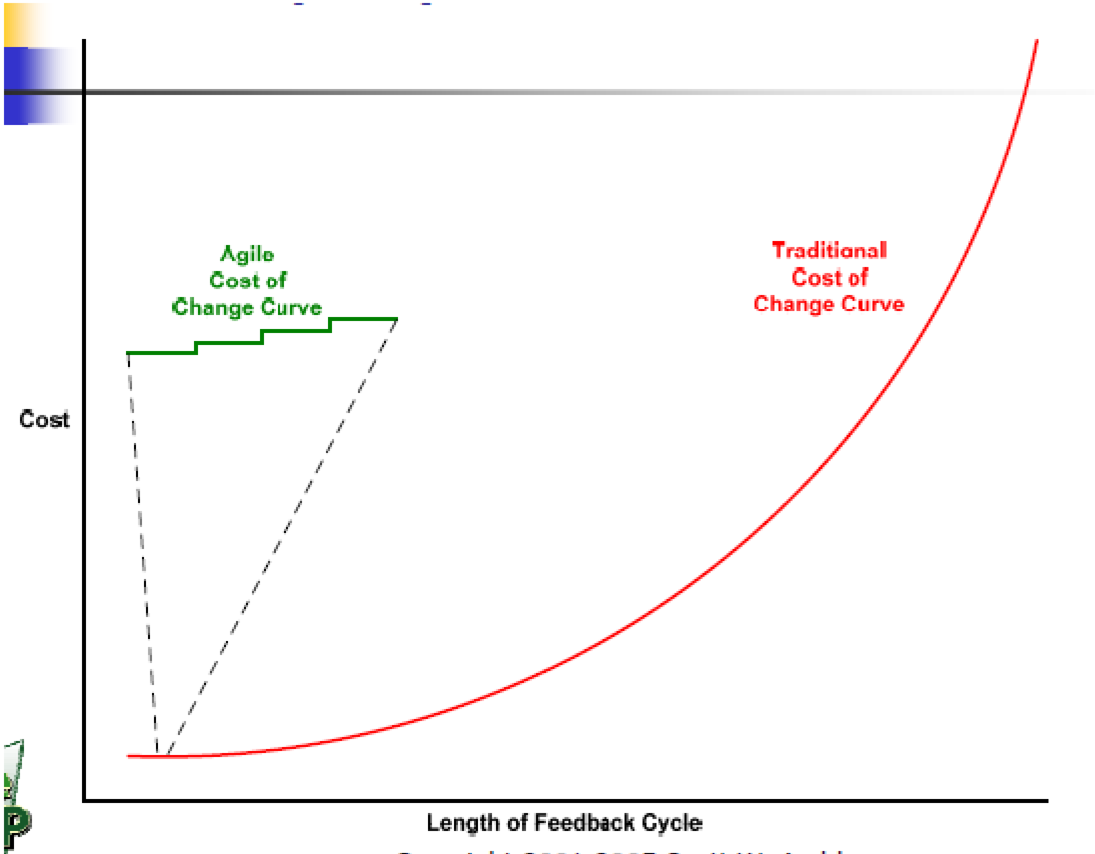
AUP BY SCOTT W.AMBLER

Agile Unified Process (AUP)

- UML is the main modeling language, but AUP is not restricted to UML.
- Use case driven
- Iterative-incremental
- Architecture-centric
- Covering the full generic lifecycle

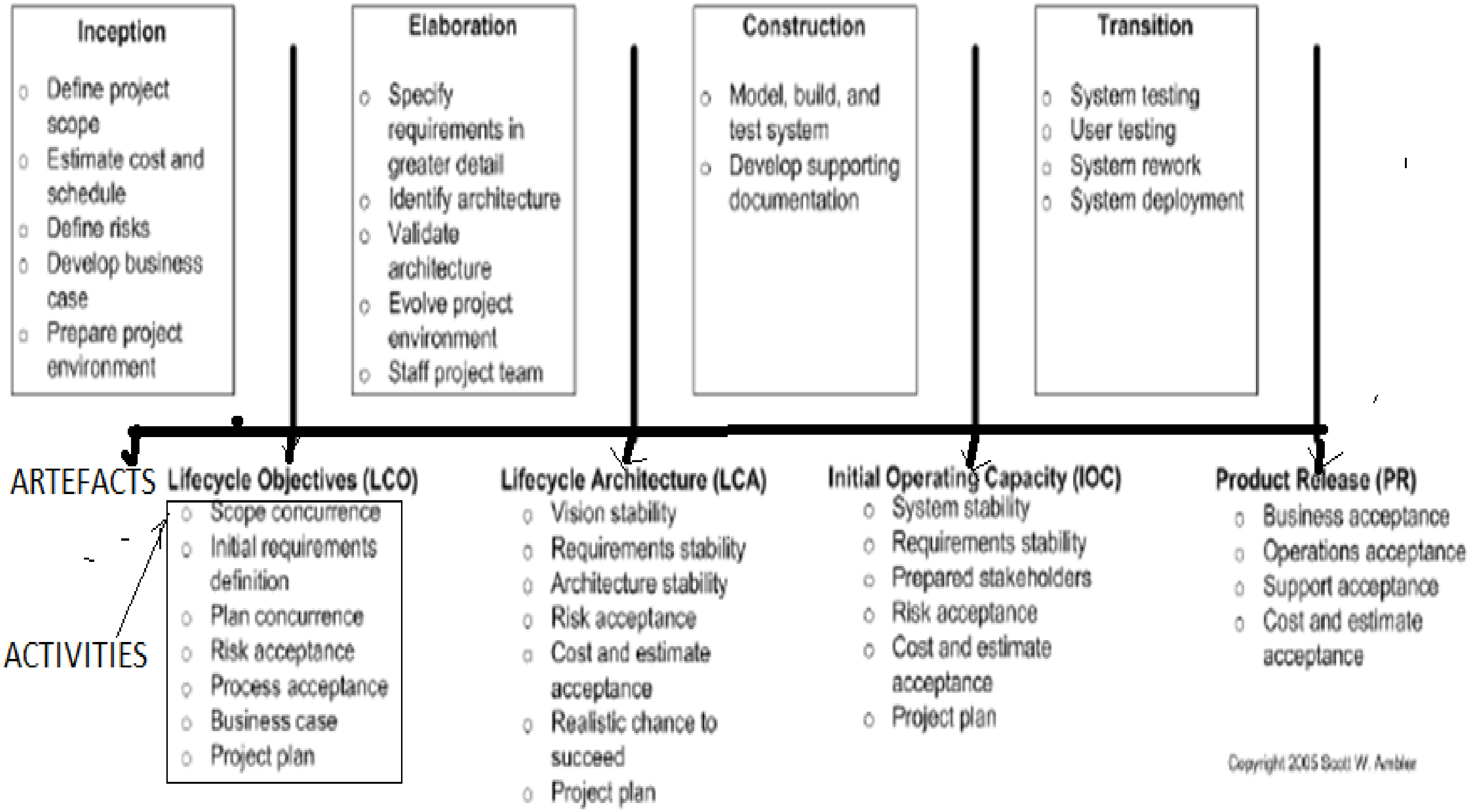
Agile UP ModeL-Just In Time(JIT)

- Observation: It’s the Same Cost Curve (JIT)



Copyright 2001-2005 Scott W. Ambler

Agile UP Phases and Milestones



Inception

- Define project scope
- Estimate cost and schedule
- Define risks
- Develop business case
- Prepare project environment

Elaboration

- Specify requirements in greater detail
- Identify architecture
- Validate architecture
- Evolve project environment
- Staff project team

Construction

- Model, build, and test system
- Develop supporting documentation

Transition

- System testing
- User testing
- System rework
- System deployment

Lifecycle Objectives (LCO)

- Scope concurrence
- Initial requirements definition
- Plan concurrence
- Risk acceptance
- Process acceptance
- Business case
- Project plan

Lifecycle Architecture (LCA)

- Vision stability
- Requirements stability
- Architecture stability
- Risk acceptance
- Cost and estimate acceptance
- Realistic chance to succeed
- Project plan

Initial Operating Capacity (IOC)

- System stability
- Requirements stability
- Prepared stakeholders
- Risk acceptance
- Cost and estimate acceptance
- Project plan

Product Release (PR)

- Business acceptance
- Operations acceptance
- Support acceptance
- Cost and estimate acceptance

ARTEFACTS

ACTIVITIES

Discipline

Model

Implementation

Test

Deployment

Configuration Management

Project Management

Environment

For each discipline, AUP defines sets of:

□ *artefacts* (work products);

□ *activities* (units of work on the artefacts);

□ *roles* (responsibilities taken on by development team members).

ARTEFACTS OF PHASES ;

LCO,LCA,IOC ,PR

AUP: Process – Phases

- Like RUP, the overall development *cycle* consists of four *phases:*
    - ***Inception:*** The goal is to identify the initial scope of the project, a potential architecture for the system, and to obtain initial project funding and stakeholder acceptance.
    - ***Elaboration:*** The goal is to prove the system architecture.
    - ***Construction:*** The goal is to build working software on a regular, incremental basis which meets the highest-priority needs of project stakeholders.
    - ***Transition:*** The goal is to validate and deploy the system into the production environment.

AUP: Process – *Inception* Phase

- Tasks include:

  1. Define project scope.
  2. Estimate cost and schedule.
  3. Define risks.
  4. Determine project feasibility.
  5. Prepare the project environment: reserving workspace for the team, requesting the people needed, obtaining the necessary hardware/software, and tailoring the AUP.

AUP: Process – *Elaboration* Phase

- Tasks include:
  1. Produce an architectural prototype for the system.
  2. Evolve the requirements model.
  3. Draft a coarse-grained project plan for the construction phase.
  4. Ensure that critical tools, processes, standards, and guidelines are put in place for the construction phase.
  5. Understand and eliminate high-priority risks of the project.

AUP: Process – *Construction* Phase

- Tasks include:

  1. Prioritize and understand the requirements.
  2. Model storm a solution.
  3. Coding and testing the software.
  4. Deploying early releases of the system to obtain user feedback.

AUP: Process – *Transition* Phase

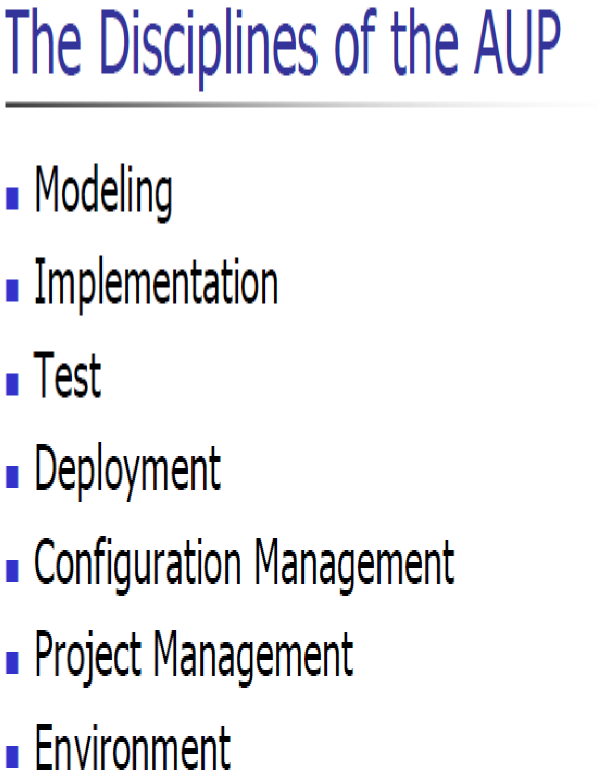
- Tasks include:
  1. Test and validate the complete system.
  2. Integrate the system with existing systems.
  3. Convert legacy databases and systems to support the new release.
  4. Train the users of the new system.
  5. Deploy the new system into production.

The Disciplines of the AUP

- Modeling
- Implementation
- Test
- Deployment
- Configuration Management
- Project Management
- Environment

AUP: Process – Iterations and Disciplines

- Each phase can be further broken down into *iterations.*
- An iteration is a complete development loop resulting in a release of an executable increment to the system.
- Each iteration consists of seven work areas (*disciplines*) performed during the iteration.
- For each discipline, AUP defines sets of:
  - *artefacts* (work products);
  - *activities* (units of work on the artefacts);
  - *roles* (responsibilities taken on by development team members).



The Disciplines of the AUP

- Modeling
- Implementation
- Test
- Deployment
- Configuration Management
- Project Management
- Environment

AUP: Process – Disciplines

1. **Model**. The goal is to understand the business of the organization, the problem domain being addressed by the project, and to identify a viable solution to address the problem domain.

2. **Implementation**. The goal is to transform the model(s) into executable code and to perform a basic level of testing, in particular unit testing.

3. **Test**. The goal is to perform an objective evaluation to ensure quality. This includes finding defects, validating that the system works as designed, and verifying that the requirements are met.

4. **Deployment**. The goal is to plan for the delivery of the system and to execute the plan to make the system available to end users.

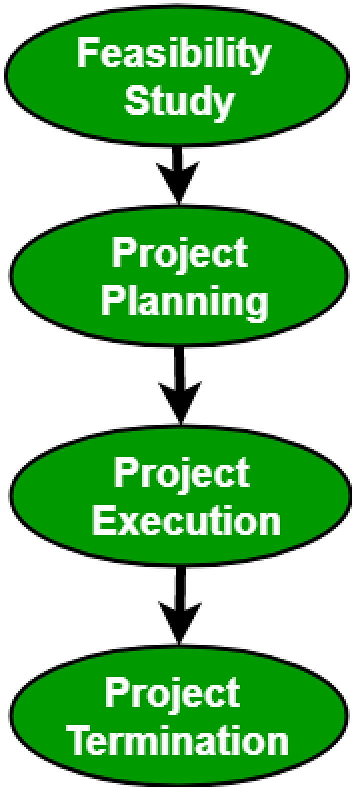
5. **Configuration Management.** The goal is to manage access to the project artifacts. This includes not only tracking artifact versions over time but also controlling and managing changes to them.

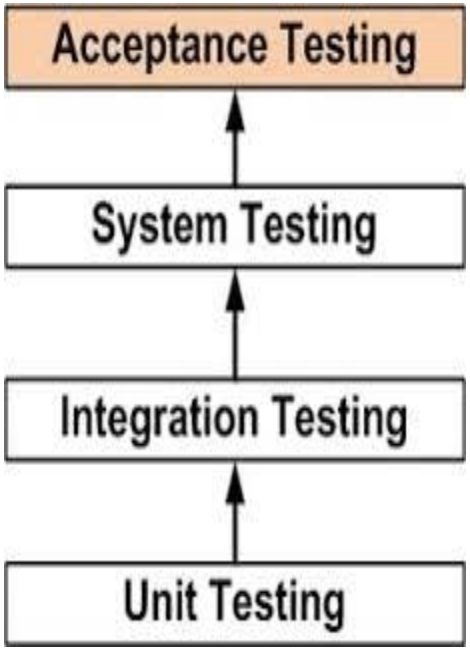
6. **Project Management.** The goal is to direct the activities that take place on the project. This includes managing risks, directing people, and coordinating with people and systems outside the scope of the project to be sure that it is delivered on time and within budget.

7. **Environment.** The goal is to support the rest of the effort by ensuring that the proper process, guidance (standards and guidelines), and tools (hardware, software, etc.) are available for the team as needed.

Project Management Process

- **Project Management Process consists of the following 4 stages:**
- Feasibility study
- Project Planning
- Project Execution
- Project Termination





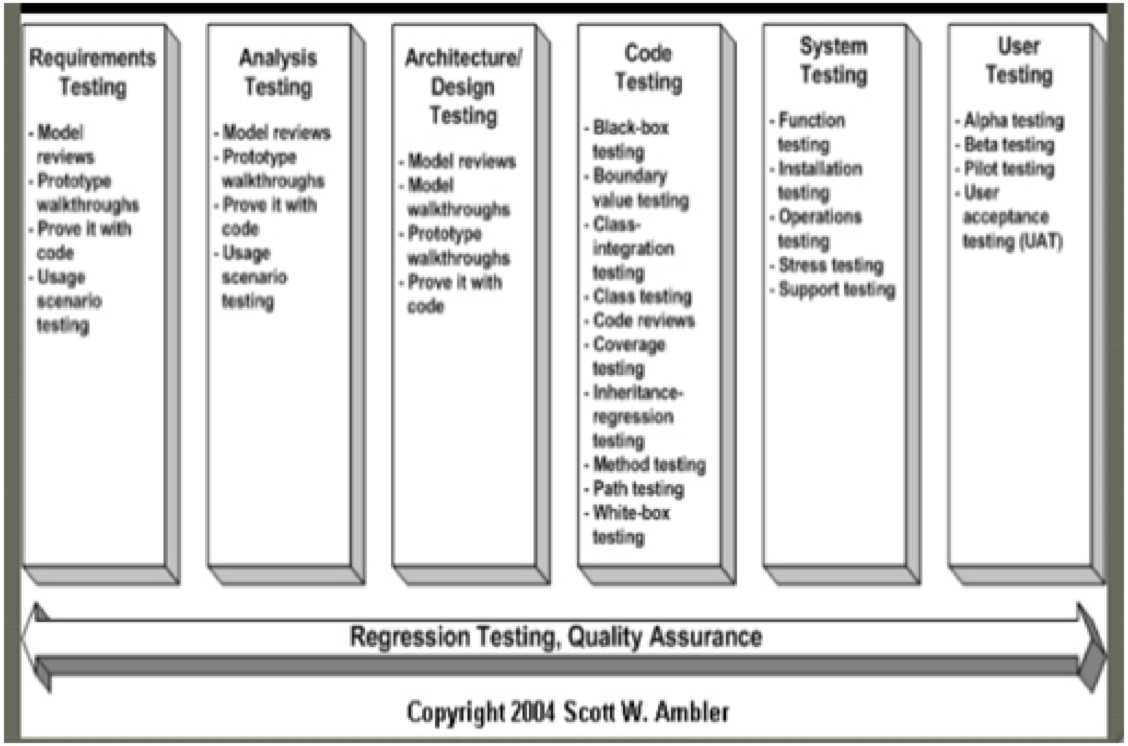
Acceptance Testing

System Testing

Integration Testing

Unit Testing

RUP Testing



Requirements Testing
- Model reviews
- Prototype walkthroughs
- Prove it with code
- Usage scenario testing

Analysis Testing
- Model reviews
- Prototype walkthroughs
- Prove it with code
- Usage scenario testing

Architecture/ Design Testing
- Model reviews
- Model walkthroughs
- Prototype walkthroughs
- Prove it with code

Code Testing
- Black-box testing
- Boundary value testing
- Class-integration testing
- Class testing
- Code reviews
- Coverage testing
- Inheritance-regression testing
- Method testing
- Path testing
- White-box testing

System Testing
- Function testing
- Installation testing
- Operations testing
- Stress testing
- Support testing

User Testing
- Alpha testing
- Beta testing
- Pilot testing
- User acceptance testing (UAT)

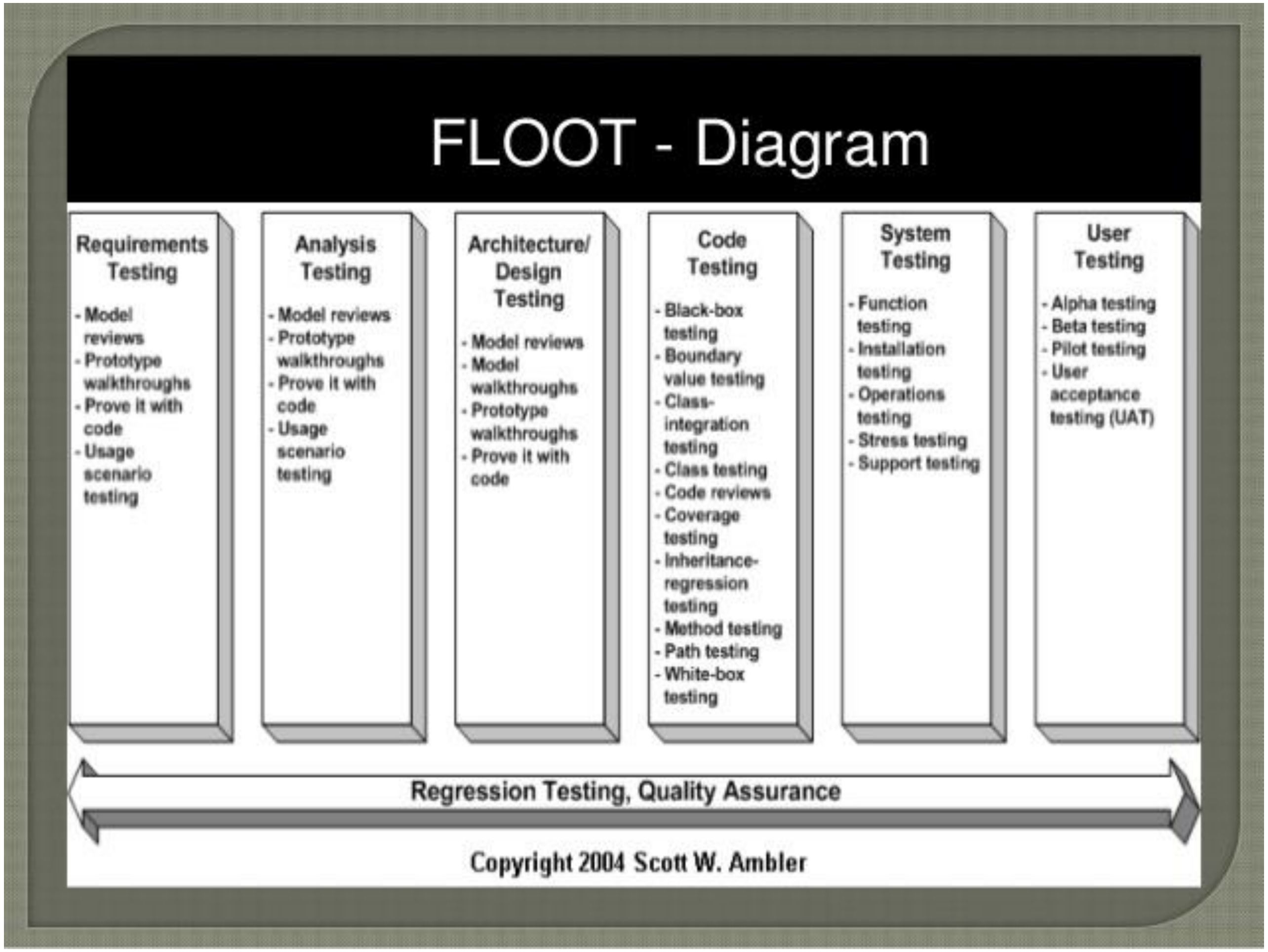
Regression Testing, Quality Assurance

Copyright 2004 Scott W. Ambler

AUP testing : Full life CYCLE OOT testing

The Full Life Cycle Object-Oriented Testing (FLOOT) Method BY ambler

The Full Life Cycle Object-Oriented Testing (FLOOT) Method



FLOOT - Diagram

Requirements Testing
- Model reviews
- Prototype walkthroughs
- Prove it with code
- Usage scenario testing

Analysis Testing
- Model reviews
- Prototype walkthroughs
- Prove it with code
- Usage scenario testing

Architecture/ Design Testing
- Model reviews
- Model walkthroughs
- Prototype walkthroughs
- Prove it with code

Code Testing
- Black-box testing
- Boundary value testing
- Class-integration testing
- Class testing
- Code reviews
- Coverage testing
- Inheritance-regression testing
- Method testing
- Path testing
- White-box testing

System Testing
- Function testing
- Installation testing
- Operations testing
- Stress testing
- Support testing

User Testing
- Alpha testing
- Beta testing
- Pilot testing
- User acceptance testing (UAT)

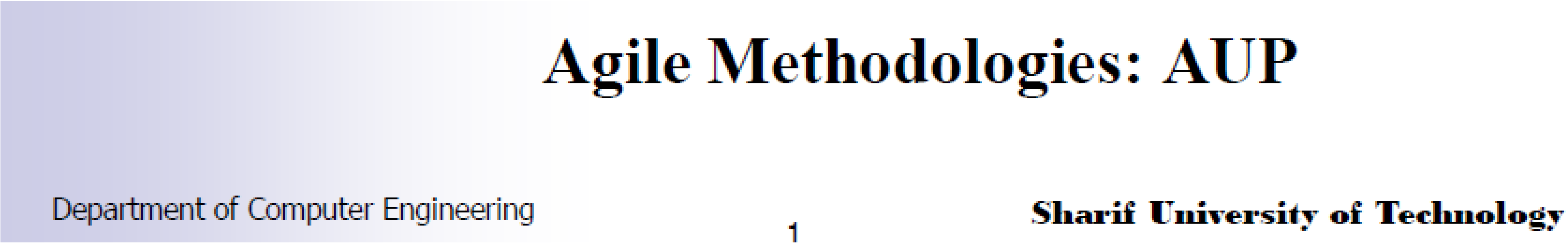
Regression Testing, Quality Assurance

Copyright 2004 Scott W. Ambler

Walkthrough in software testing is used to review documents with peers, managers, and fellow team members who are guided by the author of the document to gather feedback and reach a consensus

References

Ambler, S. W., "The Agile Unified Process (AUP)", Ambysoft Corporation, 2006, Available on the Web at: http://www.ambysoft.com/unifiedprocess/agileUP.html, Last visited in November 2017.



Agile Methodologies: AUP

Department of Computer Engineering

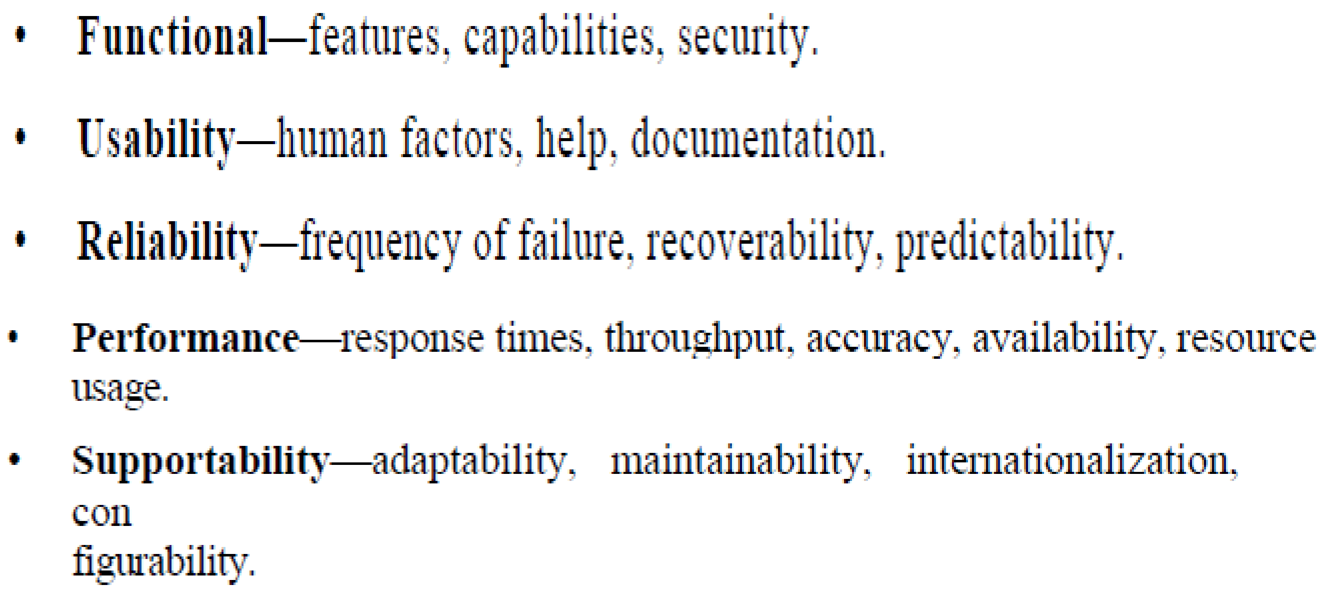
1

Sharif University of Technology

What is FURPS+ ?

Functional , Usability, Reliability, Performance, Supportability

- Types of Requirements
- In the UP, requirements are categorized according to the FURPS+ model, a useful mnemonic with the following meaning:



- **Functional**—features, capabilities, security.
- **Usability**—human factors, help, documentation.
- **Reliability**—frequency of failure, recoverability, predictability.
- **Performance**—response times, throughput, accuracy, availability, resource usage.
- **Supportability**—adaptability, maintainability, internationalization, con
  figurability.

The "+" in FURPS+ indicates ancillary and sub-factors, such as:

- **Implementation**—resource limitations, languages and tools, hardware, ...
- **Interface**—constraints imposed by interfacing with external systems.
- **Operations**—system management in its operational setting.
- **Packaging**
- **Legal**—licensing and so forth.

Some of these requirements are collectively called the **quality attributes, quality requirements,** or the "-ilities" of a system. These include usability, reliability, performance, and supportability. In common usage, requirements are categorized as **functional** (behavioral) or **non-functional** (everything else);

LEARNING NEVER ENDS…

Guide To Becoming A Self-Taught
Software Developer

https://www.youtube.com/watch?v=J6rVaFzOEP8

References

1. , **The Agile Unified Process (AUP)**

http://www.ambysoft.com/unifiedprocess/agileUP.html

2."cloud computing." Dictionary.com Unabridged. Random House, Inc. 27 Feb. 2012. <Dictionary.com http://dictionary.reference.com/browse/cloud computing>.

1. “Cloud computing security.” *Wikipedia, The Free Encyclopedia.* Wikimedia Foundation, Inc.. 20 Feb 2012. Web. 27 Feb 2012. <http://en.wikipedia.org/wiki/Cloud_computing_security>.
2. “Cloud computing.” *Wikipedia, The Free Encyclopedia.* Wikimedia Foundation, Inc.. 28 February 2012. Web. 27 Feb 2012. <http://en.wikipedia.org/wiki/Cloud_computing>.
3. nanospeak, . "Top Cloud Computing Companies List To Watch and Invest in 2012." *Hub Pages*. 10 Feb 2010: n. page. Web. 29 Feb. 2012. <http://nanospeck.hubpages.com/hub/Best-Cloud-Service-Providers>.