# Iterative, Evolutionary and Agile

Software Design and Analysis CSCI 2040

## Objectives

- Define an iterative process.
  - Introduce fundamental concepts in the unified process (UP).
- Define agile process.
- Compare waterfall vs iterative process.

#### Intro

- Agile flexible modelling practices are key to applying the UML in an effective way.
- Iterative and evolutionary development contrasts with a "waterfall " (sequential)
  - Early programming and testing of a partial system, in repeating cycles.
  - Feedback is used to clarify and improve the evolving specifications.
- The unified process is a relatively popular sample iterative method.

## Waterfall (Sequential)



## Waterfall Failure

- Success/failure studies show that the waterfall is strongly associated with the highest failure rates for software projects.
- Historically promoted due to belief or hearsay rather than statistically significant evidence..
- Research (e.g., IBM TJ Watson Research Center) demonstrates that iterative methods are associated
  - with much higher success and productivity rates, and
  - Iower defect levels.

#### **Percentage of Requirements Changes**



## **Iterative Development**

"You should use iterative development only on projects that you want to succeed.."

Martin Fowler

https://www.martinfowler.com/

Author of the book 'Refactoring: Improving the Design of Existing Code',

## Unified Process (UP)

## What is Unified Process?

- The unified process has emerged as a popular iterative software development process for building object-oriented systems.
- Since the unified process is common and promotes widely recognized best practices, it is useful for industry professionals to know it..
  - and students entering the workforce to be aware of it...

## **UP** Characteristics

- Iterative and Incremental phases are divided into a series of timeboxed iterations.
- Architecture Centric insists that architecture is at the heart of the project team's efforts to shape the system.
- Risk Focused requires the project team to focus on addressing the most critical risks early in the project life cycle.

#### Are Other Methods Complementary?

- The UP is very flexible and open
- It encourages including skillful practices from other iterative methods, such as from
  - Extreme Programming (XP),
  - Scrum, and so forth.

#### What is Extreme Programming?



#### Extreme Programming – Programming in Pairs



#### **Extreme Programming**



Copyright 3 2003 United Feature Syndicate, Inc.

#### Scrum – War Rooms





#### **Iterative and Evolutionary Development**

- A key practice in both the UP and most other modern methods is iterative development.
- Development is organized into a series of short, fixed-length (for example, three-week) mini-projects called iterations.
  - The outcome of each is a tested, integrated, and executable *partial* system.
  - Each iteration includes its own requirements analysis, design, implementation, and testing activities.

#### **Iterative and Evolutionary Development**



- The result of each iteration is an executable but incomplete system.
  - The system may not be eligible for production deployment until after many iterations, e.g., 10, 15 or 20 iterations.
- Iterative feedback and evolution leads eventually towards the desired system (not prototype!).
- The requirements and design instability lowers over time.

#### **Iterative and Incremental Development**

Early iterations are farther from the "true path" of the system. Via feedback and adaptation, the system converges towards the most appropriate requirements and design.

In late iterations, a significant change in requirements is rare, but can occur. Such late changes may give an organization a competitive business advantage.



one iteration of design, implement, integrate, and test

#### Systems Change over Time



#### Benefits to Iterative Development

- Less project failure, better productivity, and lower defect rates.
- Early rather than late mitigation of high risks
- Early visible progress
- Early feedback and user engagement
  - leading to a refined system that more closely meets the real needs of the stakeholders
- Managed complexity;
  - the team is not overwhelmed by "analysis paralysis" or very long and complex steps

#### **Evolutionary Analysis and Majority in Early Iterations**



#### **Timeboxed Iterations**

- A key idea is that iterations are timeboxed, or fixed in length.
  - For example, if the next iteration is chosen to be three weeks long, then the partial system *must* be integrated, tested, and stabilized by the scheduled date.
- Date slippage is illegal.
- Otherwise remove tasks or requirements from the iteration, and include them in a future iteration, rather than slip the completion date.

## Agile

#### **Agile Methods and Attitudes**

- Agile development methods usually
  - apply timeboxed iterative and evolutionary development,
  - employ adaptive planning,
  - promote incremental delivery
  - and include other values and practices that encourage *agility*, rapid and flexible response to change.

#### **Agile Methods and Attitudes**

- It is not possible to exactly define agile methods, as specific practices vary widely.
- They promote practices and principles that reflect an agile sensibility of simplicity, lightness, communication, self-organizing teams, and more..
  - e.g., just draw UML diagrams on the board.

#### Adhoc UML Diagrams



#### Agile Methods and Attitudes

- Example agile practices from the Scrum method include
  - a common project workroom and
  - self-organizing teams that coordinate through a daily stand-up meeting with four special questions each member answers.
- Example practices from the Extreme Programming (XP) method include
  - programming in pairs and
  - test-driven development.

#### **Agile Principles**

- www.agilealliance.com
  - https://www.agilealliance.org/agile101/12principles-behind-the-agile-manifesto/

# Agile UP

#### What is an Agile UP?

- The UP was NOT meant by its creators to be heavy or un-agile
- It was meant to be adopted and applied in the spirit of adaptability and lightnessan agile UP.
  - Prefer a *small* set of activities and artifacts.
  - A series of iterations, based on feedback.
  - Apply the UML with agile modeling practices.
  - There is not a *detailed* plan for the entire project.

#### **UP** Phases

- 1. Inception approximate vision, business case, scope, vague estimates.
- 2. Elaboration refined vision, iterative implementation of the core architecture, resolution of high risks, 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.

#### **UP Phases Diagram**



#### **Main UP Disciplines**

#### Requirements

 The Use-Case Model and Supplementary Specification artifacts to capture functional and non-functional requirements.

#### Business Modeling

The Domain Model artifact, to visualize noteworthy concepts in the application domain.

#### Design

The Design Model artifact, to design the software objects.

#### **Disciplines and Phases**

- During one iteration work goes on in most or all disciplines.
- However, the relative effort across these disciplines changes over time.
- Early iterations naturally tend to apply greater relative emphasis to requirements and design, and later ones less so.
  - This is because the requirements and core design stabilize through a process of feedback and adaptation.

#### **Disciplines and Phases**



#### How to Customize Unified Process

- Are There Optional Artifacts or Practices in the UP?
  - Yes! Almost everything is optional.
  - Some UP practices and principles are invariant, such as
    - iterative development,
    - risk-driven development,
    - continuous verification of quality.

#### Sample Case of UP Artifacts

Discipline	Artifact	Incep.	Elab.	Const.	Trans.
	Iteration-*	11	ElEn	CL.Cn	T1T2
Business Modeling	Domain Model		s		
Requirements	Use-Case Model	s	r		
	Vision	s	r		
	Supplementary Specification	s	r		
	Glossary	s	r		
Design	Design Model		s	r	
	SW Architecture Document		s		
	Data Model		s	r	
Implementation	Implementation Model		s	r	r
Project Management	SW Development Plan	s	r	r	r
Testing	Test Model		s	r	
Environment	Development Case	S	r		

### Quiz

- What is the difference between waterfall methodology and iterative process?
  - Which one and why would you recommend to use in the complex project?
- What is unified process?
- What is an example of Extreme Programming and Scrum, respectively?
- What are the main UP disciplines? Does the emphasis on disciplines change over time?
- Is UP customizable? What are the invariants of UP?

#### Actions

- Review Slides.
- Read about agile at <u>www.agilealliance.com</u>
- Read Chapter 2 (Iterative Development and The Unified Process)
  - Applying UML and Patterns, Craig Larman
- Optional
  - Steve Jobs' Stanford Commencement Speech
  - https://www.youtube.com/watch?v=UF8uR6Z6KLc