CSE291 - Introduction To Software Engineering
(Fall 2018)

Lecture 3

RAD and Agile Software Development

Outline

Rapid Software Development

- RAD

Agile Software Development

- Extreme Programming

Rapid Application Development

- Rapid development and delivery is now often the most important requirement for software systems –why?
- Businesses now operate in a rapidly changing environment – they have to respond to new opportunities
- Software has to evolve quickly to reflect changing business needs.

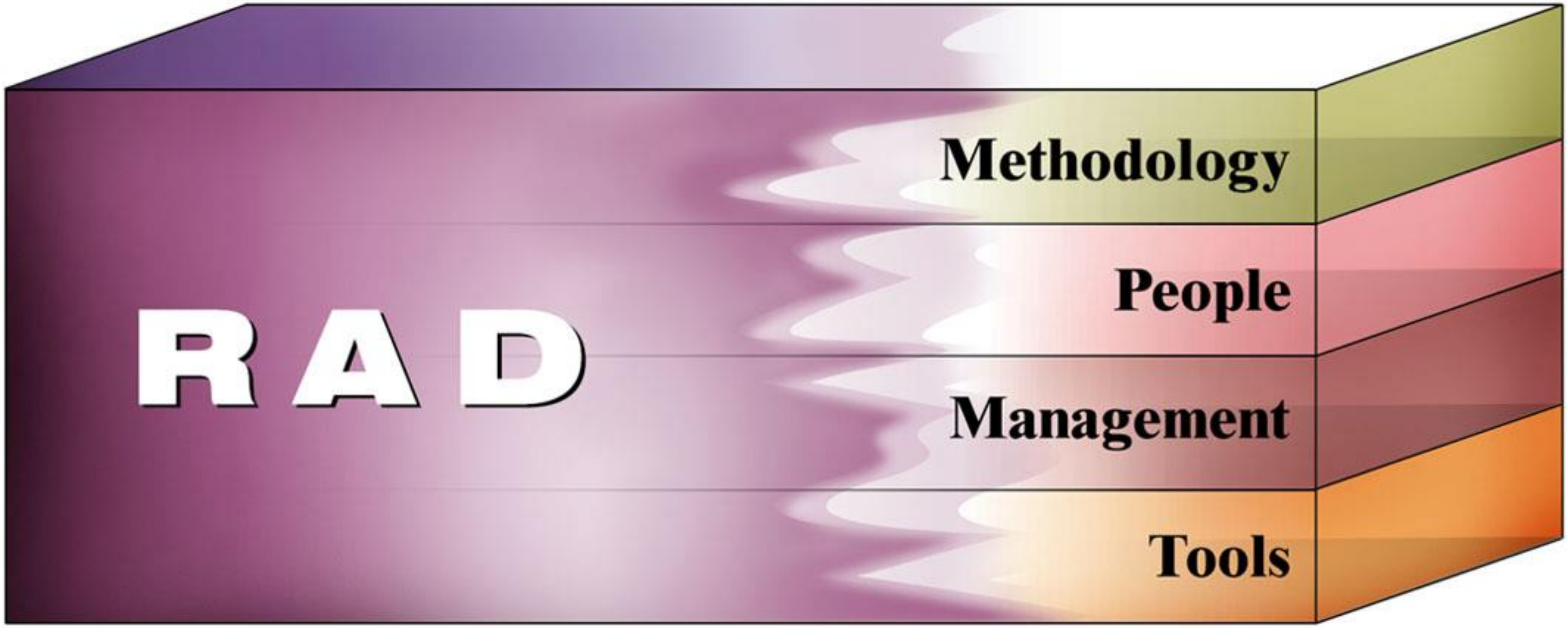
RAD (Rapid Application Development)

*“Rapid Application Development (RAD) is a development lifecycle designed to give **much faster development** and **higher-quality results** than those achieved with the traditional lifecycle.”*

RAD(Rapid Application Development)

- RAD takes advantage of **automated tools and techniques**.
- RAD replaces hand-design and coding processes, which are dependent upon the skills of isolated individuals, with automated design and coding
- The availability of powerful **CASE** software makes it possible for developers to create systems much faster than ever before.

Essential Aspects of RAD



RAD(Rapid Application Development)

METHODOLOGY

- Combining the best available techniques
  - Using evolutionary prototypes
  - Using workshops instead of interviews →for requirements and review design
  - Selecting a set of CASE tools for modeling, prototyping and code reusability

Structure of RAD Life Cycle

Requirement Planning – Concept definition Stage

- Defines Business functions /Requirements
- Determine the system's scope.

User Design – Functional Design Stage

- uses workshops
  - To model the system's data and processes
  - To build a working prototype of critical system components.

Construction – Development Stage

- completes the construction of application system

Implementation – Deployment Stage

- final user testing and Training
- implementation of the application system

RAD(Rapid Application Development)

PEOPLE

- Success is dependent upon the involvement of people with the right skills and talents
- Must be carefully selected, highly trained, and highly motivated.
- At the Requirements Planning and User Design stages, key end users must be available to participate in workshops.
- While the system is being constructed, the Construction Team, which uses the CASE toolset must be available
- At the end of the development cycle, the Team, which handles training, must also be ready to move quickly.

RAD(Rapid Application Development)

The key players in a Rapid Application Development project include:

Sponsor

- A high-level user executive who funds the system.

User Coordinator

- A user appointed by the Sponsor to oversee the project from the user perspective.

Requirements Planning Team

- A team of high-level users who participate in the Joint requirements Planning workshop.

User Design Team

- A team of users who participate in the design workshop.

RAD(Rapid Application Development)

User review Board

- A team of users who review the system after construction and decide whether modifications are necessary.

Training Manager

- The person responsible for training users to work with the new system.

Project Manager

- The person who oversees the development effort.

Construction Team

- The SWAT (Skilled Workers with Advanced Tools) Team is a small team of two to six developers who are highly trained to work together at high speed.
- Skilled in the RAD methodology and in using the chosen CASE toolset.

Workshop Leader

- The specialist who organizes and conducts the workshops for Joint Requirements Planning and Joint Application Design.

RAD(Rapid Application Development)

MANAGEMENT

- Management must be totally committed to RAD in order to manage the change in culture.

- Management must pay careful attention to human motivation.

- **Managers should target ‘Early Adapters.’**

- Managers must remember the importance of comprehensive and quality training in the use of tools.

RAD(Rapid Application Development)

TOOLS

- The RAD methodology uses both computerized tools and human techniques to achieve the goals of high-speed and high quality.

- The power tools utilized in Rapid Application Development are Computer-Aided Systems Engineering (CASE) tools.

- A fundamental principle of RAD tools is that diagrams are employed whenever possible as an aid to clear thinking.

Agile Software Development

An Agile Process

- Agile means being able to “Deliver quickly. Change quickly. Change often”
- Develops software iteratively
- Delivers multiple ‘software increments’
- Adapts as changes occur

Agile Software Development

Characteristics of Agile Software Development

- Small to medium sized teams
- changing requirements
- changing techniques
- Simple design

The Manifesto for Agile Software Development

“We are uncovering better ways of developing software by doing it and helping others to do it. Through this work we have come to value:

- *Working software* over comprehensive documentation
- *Customer collaboration*
- *Responding to change* over following a plan

Agile Methods

- These methods
  - Focus on the code
  - Are based on an iterative approach to software development
  - Are intended to deliver working software quickly and evolve this quickly to meet changing requirements.

- The aim of agile methods is to reduce overheads in the software process (e.g. by **limiting documentation**) and to be able to respond quickly to changing requirements without excessive rework.

The Principles Of Agile Methods

Principle	Description
Customer involvement	Customers should be closely involved throughout the development process. Their role is provide and prioritize new system requirements and to evaluate the iterations of the system.
Incremental delivery	The software is developed in increments with the customer specifying the requirements to be included in each increment.
People not process	The skills of the development team should be recognized . Team members should be left to develop their own ways of working.
Adapt change	Expect the system requirements to change and so design the system to accommodate these changes.
Maintain simplicity	Focus on simplicity in both the software being developed and in the development process. Wherever possible, actively work to eliminate complexity from the system.

Existing Agile Methods

- Extreme Programming (“XP”)
- Agile Unified Process….etc

Extreme Programming

- Perhaps the best-known and most widely used agile method.
  - New versions may be built several times per day
  - Increments are delivered to customers every 2 weeks
  - All tests must be run for every build and the build is only accepted if tests run successfully.

Extreme Programming (XP)

XP Values

- Communication(Be together with your customer and fellow programmers, and talk to each other)
- Simplicity (Use simple design and programming practices, and simple methods of planning, tracking, and reporting)
- Feedback(Test your program and your practices, using feedback to steer the project)

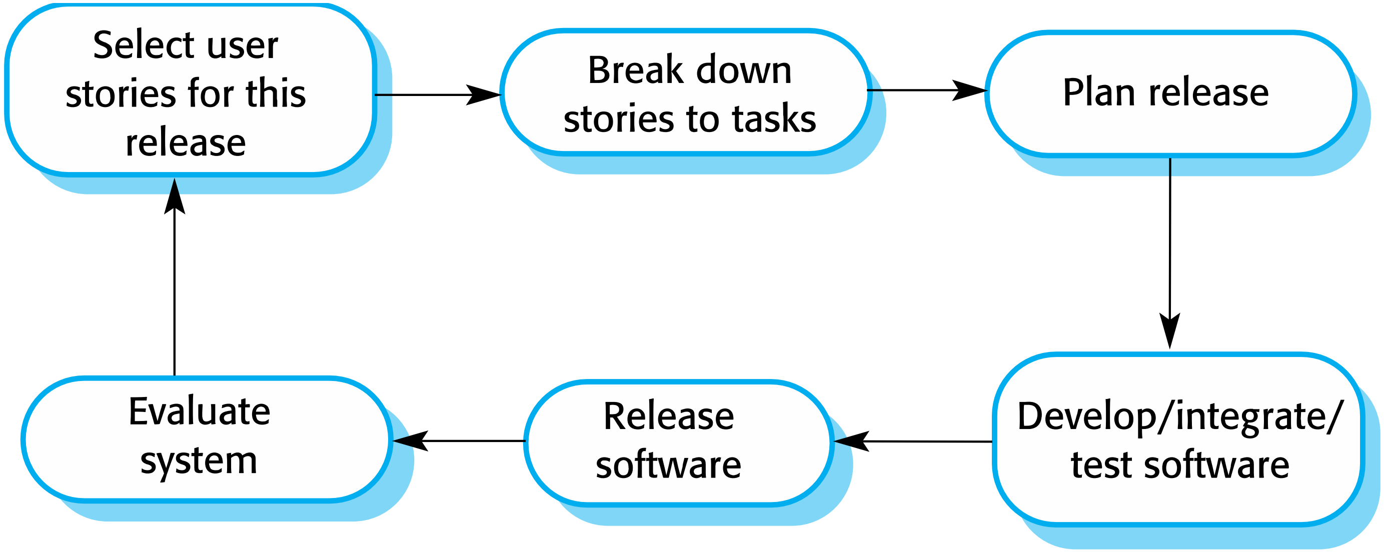
Extreme Programming Core Practices (A)

Principle or practice	Description
Incremental planning	• **Requirements are recorded on story cards** •The developers break these stories into development 'Tasks'.
Small releases	•The minimal useful set of functionality that provides business value is developed first. •Releases of the system are frequent and incrementally add functionality to the first release.
Simple design	Enough design is carried out to meet the current requirements and no more.
Test-first development	An automated unit test framework is used to write tests for a new piece of functionality before that functionality itself is implemented.
Refactoring	All developers are expected to refactor / restructure the code continuously as soon as possible code improvements are found. This keeps the code simple and maintainable.

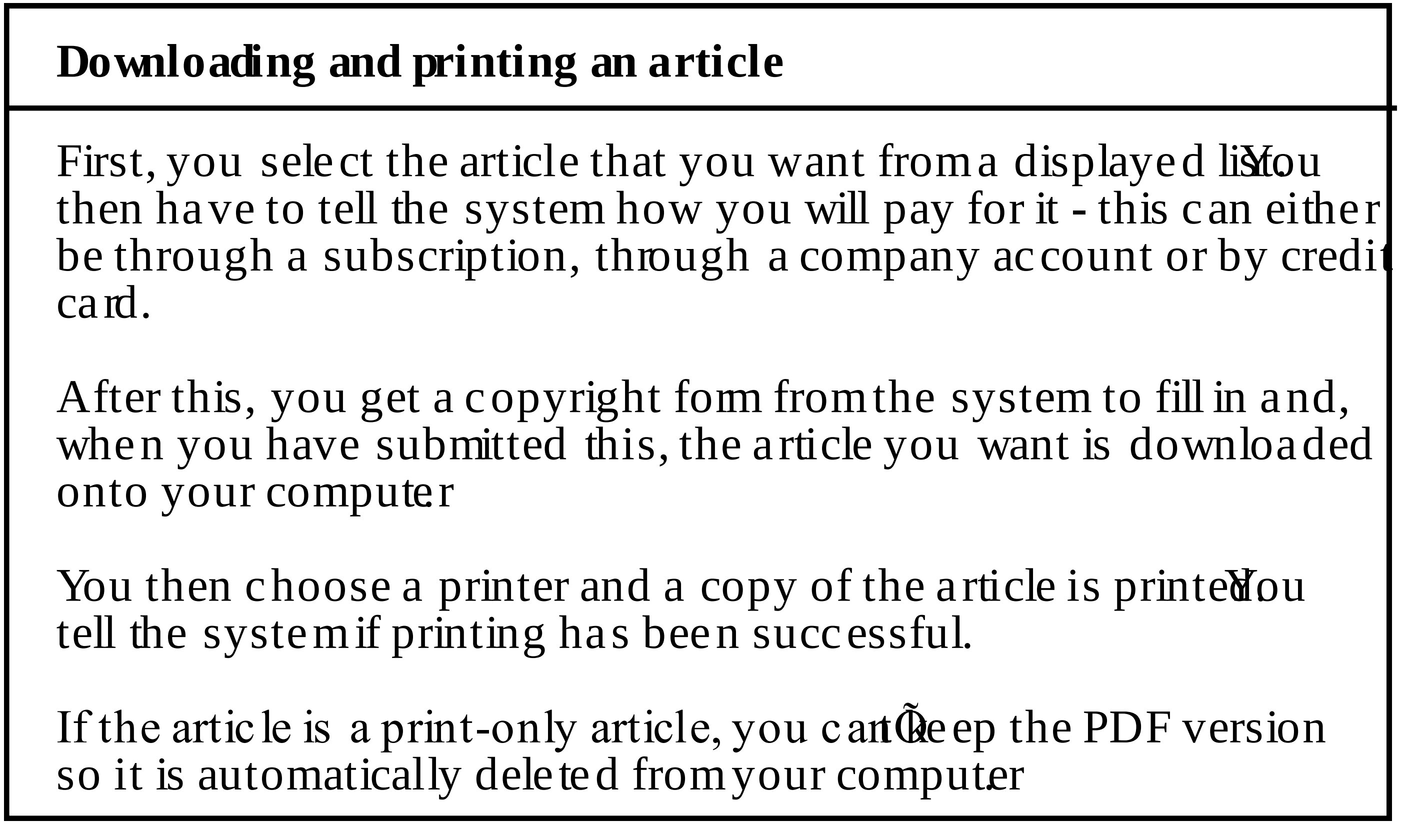
Extreme Programming Core Practices (B)

Pair programming	Developers work in pairs, checking each other’s work and providing the support to always do a good job.
Collective ownership	The pairs of developers work on all areas of the system, all the developers take responsibility for all of the code. Anyone can change anything.
Continuous integration	As soon as the work on a task is complete, it is integrated into the whole system. After any such integration, all the unit tests in the system must pass.
Sustainable pace	Large amounts of overtime are not considered acceptable as the net effect is often to reduce code quality and productivity
On-site customer	A representative of the end-user of the system (the customer) should be available full time. In an extreme programming process, the customer is a member of the development team and is responsible for bringing system requirements to the team for implementation.

The Extreme Programming Release Cycle


Select user stories for this release
Break down stories to tasks
Plan release
Develop/integrate/ test software
Release software
Evaluate system

Example – Story Card for downloading document



Downloading and printing an article

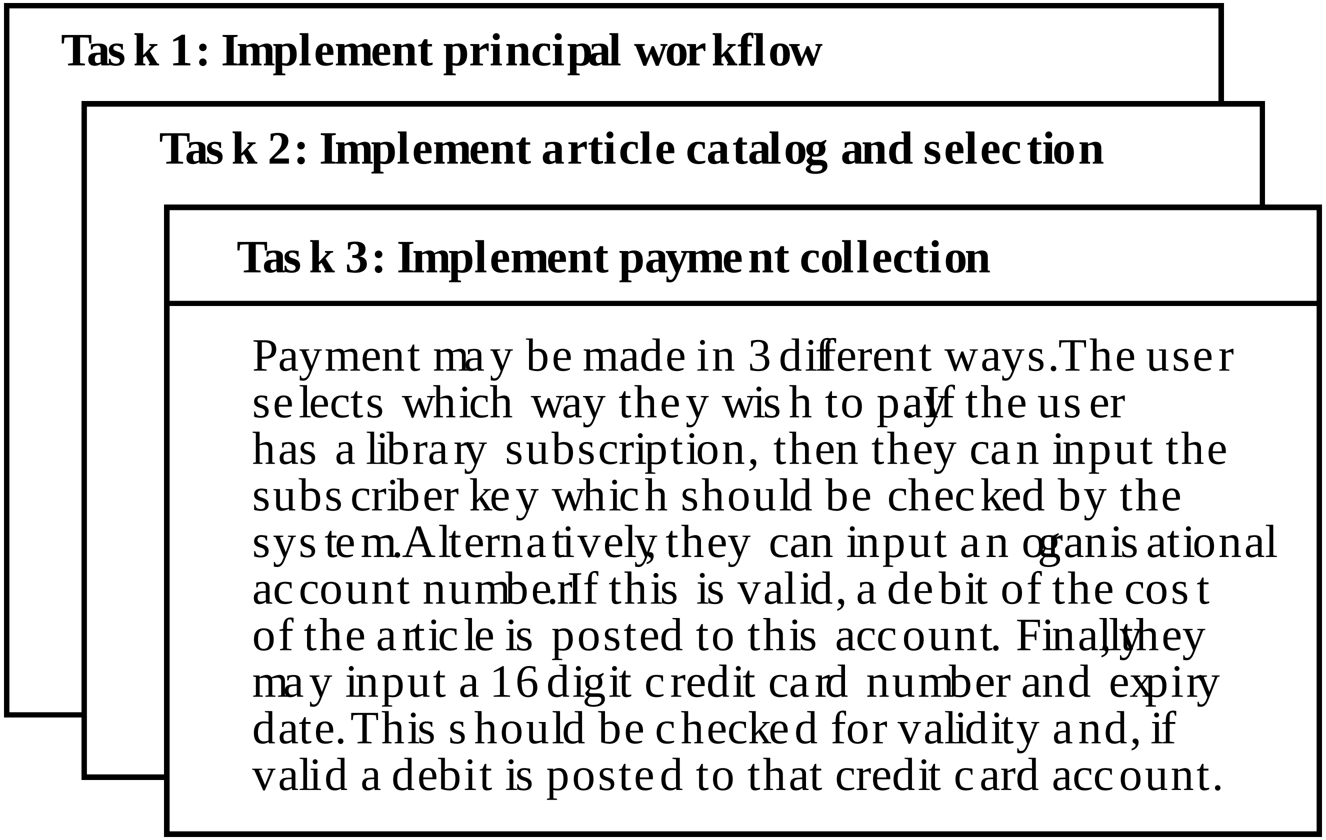
First, you select the article that you want from a displayed list. You then have to tell the system how you will pay for it - this can either be through a subscription, through a company account or by credit card.

After this, you get a copyright form from the system to fill in and, when you have submitted this, the article you want is downloaded onto your computer.

You then choose a printer and a copy of the article is printed. You tell the system if printing has been successful.

If the article is a print-only article, you canÕt keep the PDF version so it is automatically deleted from your computer.

Task Cards



Task 1: Implement principal workflow

Task 2: Implement article catalog and selection

Task 3: Implement payment collection

Payment may be made in 3 different ways. The user selects which way they wish to pay. If the user has a library subscription, then they can input the subscriber key which should be checked by the system. Alternatively, they can input an organisational account number. If this is valid, a debit of the cost of the article is posted to this account. Finally, they may input a 16 digit credit card number and expiry date. This should be checked for validity and, if valid a debit is posted to that credit card account.

Agile method Applicability

- Product development where a software company is developing a small or medium-sized product for sale.
- Custom system development within an organization, where there is a clear commitment from the customer to become involved in the development process

Agile VS RAD

- RAD are based in designing prototypes and then reengineering all the prototypes while this is mostly not occur in Agile
- RAD is managed by a project manager however in case of Agile development team members are independent.
- Agile development does continuous integration (changing environment in daily basis) and developer’s team is ready to face change but this is not applicable for RAD