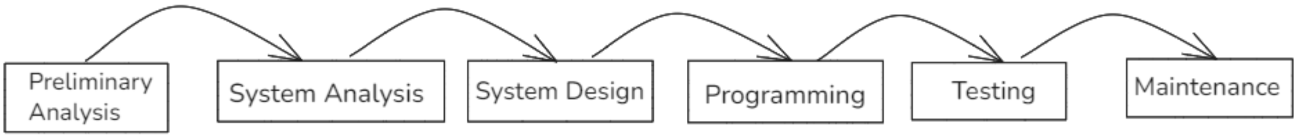
IS Course 3 - Development of Information Systems

1 - Systems Development Methodologies

Methodology	Overall Flow	Analysis	Design	Development	Testing
SDLC	Structured	Methodical	Methodical	Sequential	FD
RAD	Structured	Methodical	Interactive	Interactive	CWD
Agile	Incremental	Iterative	Iterative	Iterative	CWD
Lean	Minimalist	Iterative	Iterative	Iterative	CWD

FD = Follow Development
CWD = Combined with Development

1-1 SDLC Systems Development Life Cycle



A- Preliminary Analysis:

- Evaluation of a request to replace or create a new system.
- Technical, economic, and legal feasibility study.
- Conducted by a Requirements Analyst or Business Analyst.

B- System Analysis:

- Determination of specific requirements for the new system by one or more systems analysts.
- Documentation of procedures, interviews with key stakeholders, development of data requirements.
- Drafting of a system requirements document.
- Conducted by a system analyst.

C- System Design:

- Development of the specific technical details required for the system by a designer.
- Translation of business requirements into specific technical requirements.
- Development of the design of the user interface, database, data inputs and outputs, and reports.
- Drafting of a system design document.
- Executed by a systems analyst, developer, or systems architect.

D- Programming:

- Writing of code by programmers using the system design document as a guide.
- Development of an initial work program that meets the requirements specified in the system analysis phase and the design developed in the system design phase.
- Executed by developers, software engineers, programmers, or coders

E- Testing:

- Implementation of a series of structured tests for the software program developed during the programming phase.
- Unit testing to evaluate the various parts of the code to detect errors or bugs.
- System testing to ensure that the various system components work together correctly.
- User acceptance testing to allow users to test the system to ensure it meets their standards.
- Resolution of bugs, errors, or problems detected during testing.
- Performed by testers, test analysts, or quality assurance managers.

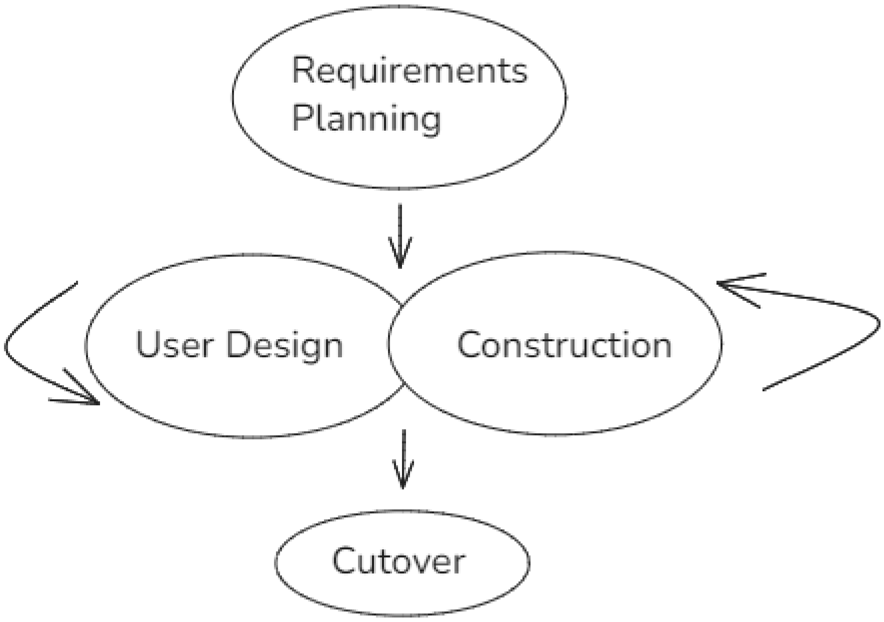
F- Deployment:

- Implementation of the new system within the organization.
- User training.
- Provision of documentation.
- Conversion of data from the old system to the new system.
- Performed by project managers, deployment managers, or implementation managers.

G- Maintenance:

- Establishment of a structured system support process.
- Correction of reported bugs.
- Evaluation and implementation of new feature requests.
- Performance of system updates and software backups for each new program release.
- Performed by maintenance managers, reliability managers, or technical support managers.

1-2 Rapid Application Development: RAD



- Focuses on quickly creating a working model of the software, obtaining user feedback, and using that feedback to update the working model.

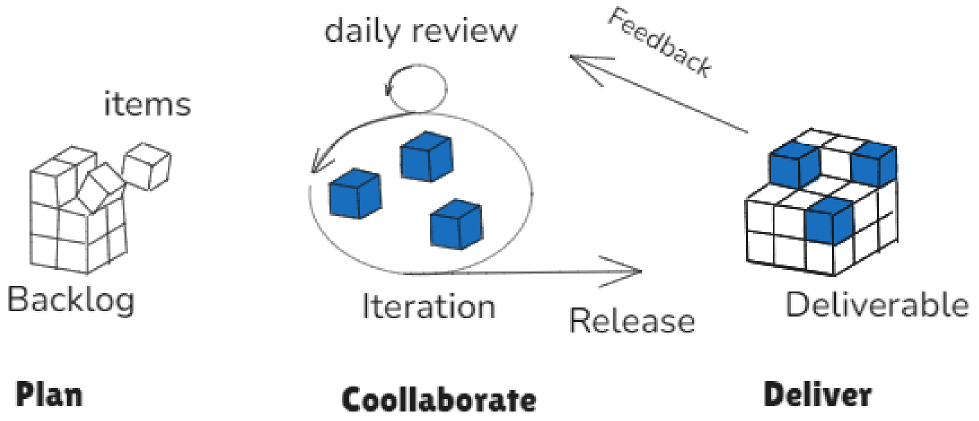
A- RAD Methodology:

- **Requirements Planning**: Defines the overall system requirements, identifies a team, and determines feasibility.
- **User Design**: Interactively creates the system design with user representatives.
- **Build**: Builds the next version of the system through an interactive process with users.
- **Transfer**: Involves the transition from the old system to the new software.

B- RAD Vs. SDLC

- RAD is better suited for small projects, while SDLC is better suited for large, resource-intensive projects.

1-3 Agile Methods

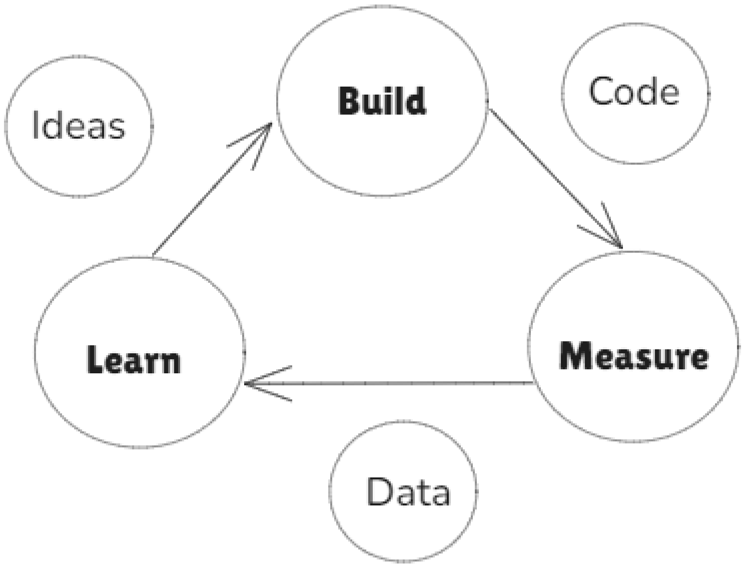


Agile Methodology Management: Iteration

Agile Methodology

- A set of methodologies that utilize incremental changes with a focus on quality and attention to detail.
- Each increment is released within a set period (called a timebox), creating a regular release schedule with very specific goals.
- Although considered a separate methodology from RAD, the two methodologies share some of the same principles, such as iterative development, user interaction, and flexibility for change.
- Agile methodologies are based on the "Agile Manifesto," first published in 2001.
- Agile development is iterative.
- Characteristics of agile methodology include:
- Small, cross-functional teams that include members of the development team and users;
- Daily status meetings to discuss the current status of the project;
- Short increments (from a few days to one or two weeks) for each change to be completed;
- A project at the end of each iteration that demonstrates progress to stakeholders.
- The goal of agile methodologies is to provide the flexibility of an iterative approach while ensuring a quality product.

1-4 Lean



Lean Methodology

- A new methodology derived from Eric Reis's best-selling book, The Lean Startup.
- The methodology focuses on creating a minimum viable product (MVP), a functional software application with just enough features to demonstrate the idea behind the project.
- The development team gives the MVP to potential users for evaluation.
- Feedback on the MVP is generated in two forms:
- direct observation and discussion with users,
- usage statistics collected from the software itself.
- Using these two forms of feedback, the team determines whether to continue in the same direction or rethink the basic idea of the project, change features, and create a new MVP.
- This change in strategy is called a pivot.
- Multiple iterations of the MVP are developed, with new features added each time based on feedback, until a final product is completed.
- The biggest difference between iterative and non-iterative methodologies is that the complete set of system requirements is not known at the start of the project.
- As each iteration of the project is released, the collected statistics and feedback are used to determine the requirements.
- The Lean methodology works best in an entrepreneurial environment where a company wants to know if software development is necessary.

2- Programming Tools

2-1 IDE: Integrated Development Environment

- A software tool that provides a set of tools to help programmers develop software.
- The key features of an IDE include:

- A code editor for writing and editing source code.
- A help system for accessing documentation about the programming language.
- A compiler or interpreter for converting source code into executable machine language.
- A debugging tool for identifying and correcting errors in the code.
- A check-in/check-out mechanism that allows teams of programmers to work simultaneously on a project without overwriting each other's code.
- Examples of IDEs
- Microsoft's Visual Studio: Supports various Microsoft programming languages, including Visual Basic, Visual C, and Visual C#.
- Oracle Eclipse: Supports a variety of programming languages, including Java, C, C++, Perl, Python, and R.

2-2 Case Tools

- While an IDE provides several tools to help the programmer write the program, the code still needs to be written.
- Computer-Aided Software Engineering (CASE) tools allow a designer to develop software with little or no programming.
- Instead, the CASE tool writes the code for the designer. CASE tools come in many varieties.
- Their goal is to generate quality code based on input created by the designer.

3- Implementation Methodologies

- Once a new system is developed or purchased, the organization must determine the best implementation method.
- Convincing a group of people to learn and use a new system can be a very difficult process.
- Asking employees to use new software and follow a new business process can have profound effects within the organization.
- There are several different methodologies an organization can adopt to implement a new system.
- Four of the most popular are listed below.

3-1 Direct Cutover:

- The organization selects a specific date on which the old system will no longer be used.
- On this date, users begin using the new system, and the old system is unavailable.
- The advantages of using this methodology are that it is quick and the least expensive.

3-2 Pilot:

- The new system is implemented in a pilot department or division before being rolled out across the organization.
- This allows the organization to identify and resolve potential issues before the new system is rolled out to a larger number of users.
- The advantages of using this methodology are that it reduces risk and allows the organization to test the new system in a live environment.

3-3 Parallel Conversion:

- The old and new systems are run in parallel during a transition period.
- This allows users to familiarize themselves with the new system before it becomes the production system.
- The advantages of using this methodology are that it reduces the risk of failure and allows users to learn the new system at their own pace.

3-4 Phased Implementation

- The features of the new system are gradually implemented (used) while the corresponding features in the old system are deactivated.
- This approach allows the organization to slowly transition from one system to another.

4- Change Management and Maintenance

4-1 Change Management

- When rolling out new systems and phasing out legacy systems, it becomes important to manage how the change is implemented within the organization.
- Change should never be introduced in a vacuum. The organization must ensure that proposed changes are communicated before they occur and plan to minimize the impact of the change that will occur after implementation.
- Change management is a critical component of IT oversight.

4-2 Maintenance

- Once a new system has been introduced, it enters the maintenance phase.
- The system is in production and is being used by the organization.
- Although the system is no longer actively developed, changes must be made when bugs are detected or new features are requested.
- During the maintenance phase, IT management must ensure that the system continues to remain aligned with business priorities and continues to function properly.

Bibliographic References

Bourgeois, David T., James L. Smith, Shouhong Wang, and Joseph Mortati. *Information systems for business and beyond*. Saylor Academy, 2019