4-Selection of an Appropriate Project Approach



Step 3 of Stepwise Project Planning

Analyse Project Characteristics



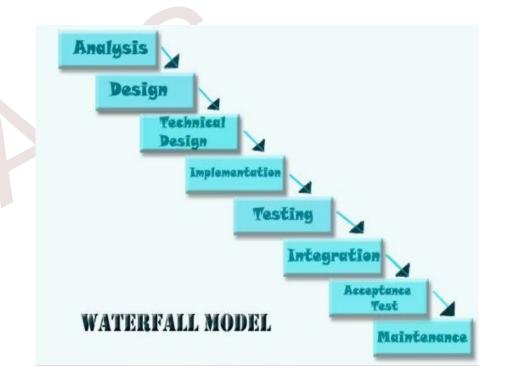
Learning Objectives

- Evaluate situations where software applications could be acquired off-the-shelf rather than being built specially
- Take account of the characteristics of the system to be developed when planning a project
- Select an appropriate process model
- Make best use of the waterfall process model where appropriate
- Reduce some risks by the creation of appropriate prototypes
- Reduce other risks by implementing the project in increments
- Identify where unnecessary organizational obstacles can be removed by using agile development methods



Waterfall Model

- It was put forth by Winston Royce in 1970
- Works well when quality is more important than Cost or Schedule
- Requirements should be known upfront
- Little opportunity for customer to preview the product
- Provides structure to inexperienced staff
- Good for new version of an existing product





Evolutionary delivery: Prototyping

'An iterative process of creating quickly and inexpensively live and working models to test out requirements and assumptions'

Main types

- 'throw away' prototypes
- evolutionary prototypes

what is being prototyped?

- human-computer interface
- functionality



Reasons for prototyping

- learning by doing
- improved communication
- improved user involvement
- a feedback loop is established
- reduces the need for documentation
- reduces maintenance costs i.e. changes after the application goes live
- prototype can be used for producing expected results



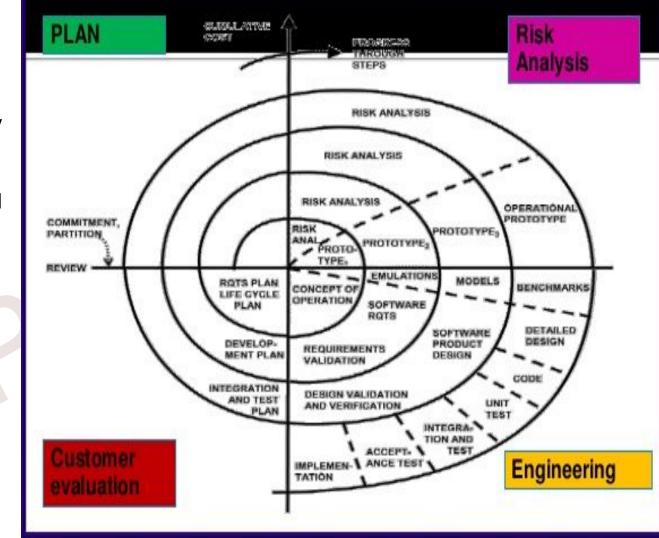
Other ways of categorizing prototyping

- Prototyping to what extent...
 - Mock-ups-user can see but can't use
 - Simulated interaction-can use with dummy data
 - Partial working models:
 - vertical some features are fully prototyped
 - Horizontal All features are there but not in detail
 - Controlling changes during Prototyping changes are categorised as
 - Cosmetic Changes
 - Local Changes
 - Global Changes



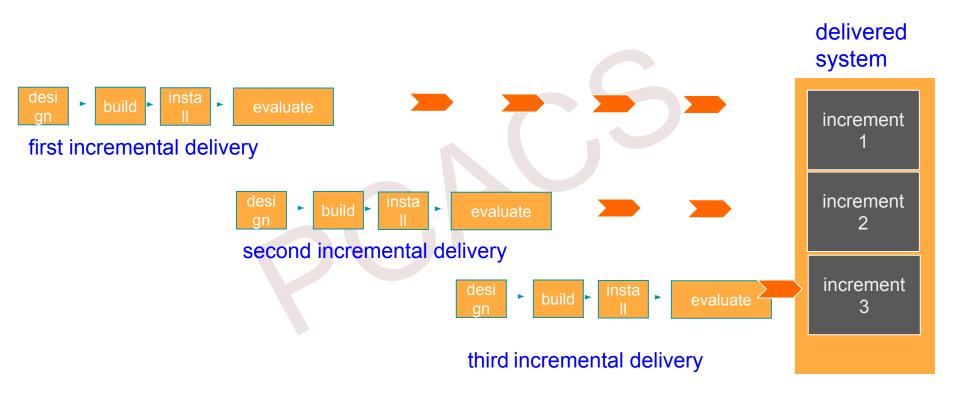
Spiral model

- 1. It was defined in 1988 by Barry Boehm
- Good for large, expensive & complicated projects
- 3. It combines features of Prototype and Waterfall
- It is divided into number of activities called Task Regions
- 5. Changing requirements can be accomodated
- 6. Users can see the product early



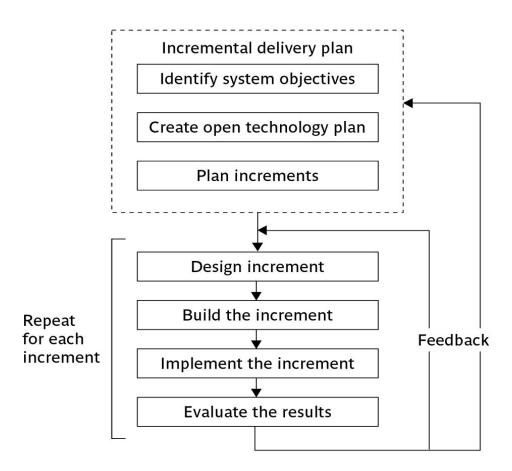


Incremental delivery





The incremental process





Incremental approach: benefits

- feedback from early stages used in developing latter stages
- shorter development thresholds
- user gets some benefits earlier
- project may be put aside temporarily
- reduces 'gold-plating'

BUT there are some possible disadvantages

- loss of economy of scale
- 'software breakage'



Overview of incremental plan

- steps ideally 1% to 5% of the total project
- non-computer steps should be included
- ideal if a step takes one month or less:
 - not more than three months
- each step should deliver some benefit to the user
- some steps will be physically dependent on others



Which step first?

- some steps will be pre-requisite because of physical dependencies
- others may be in any order
- value to cost ratios may be used
 - V/C where
 - V is a score 1-10 representing value to customer
 - C is a score 0-10 representing cost to developers



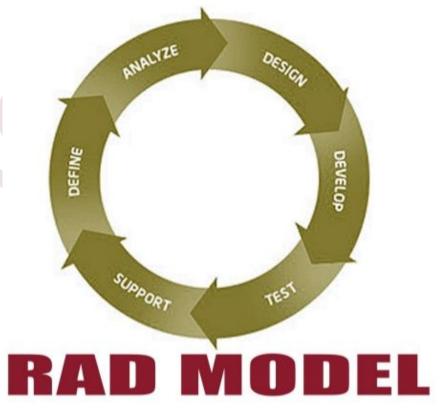
V/C ratios: an example

step	value	cost	ratio	
profit reports	9	1	9	2nd
online database	1	9	0.11	5th
ad hoc enquiry	5	5	1	4th
purchasing plans	9	4	2.25	3rd
profit- based pay for managers	9	0	inf	1st



RAD (RAPID APPLICATION DEVELOPMENT)

- It is combination of Prototyping and Incremental
- Major aim is to
 - Decrease the TIME taken & the COST of software development
 - To decrease the cost of accommodating the change request by incorporating change as early as possible
 - Develop software as early as possible
- Development takes place in series of iterations
- It focuses on CODE REUSE for faster development





Steps of Project Analysis

- 1. Identify project as **Objective Driven or Product Driven**
- 2. **Data oriented or Process Oriented**: Data oriented Inventory systems; Substantial Database is required; Process ORiented: Embedded System; Object Oriented approach is suitable.
- 3. General Tool or Application Specific
- 4. Does it involve concurrent processing
- 5. Will the system to be created be **knowledge based**
- 6. Will it require heavy use of **graphics**
- 7. Is it **safety critical** (its failure might endanger the human life)
- 8. **Hardware / Software environment** in which the system will operate (when the customer environment is different than the developers environment; like a standalone desktop would require different approach than a Mainframe / client-server environment)
- 9. Identify high level **PROJECT RISK**: greater the **uncertainties** at the beginning greater the RISK of unsuccessful ending. Different types of uncertainties are :
 - a. Product Uncertainty how well the requirements understood
 - b. Process UNcertainty when a new process is used
 - c. Resources Uncertainty like unavailability of staff (human resource)



Steps of Project Analysis (contd.)

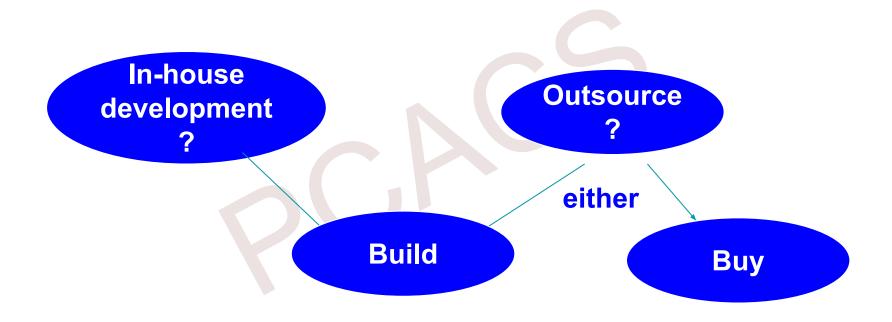
10. Take into account user requirements concerning implementation: like a specific vendor's tools/products

11. SELECTING LIFE CYCLE APPROACH

- Control Systems (Real time) Petri nets(A Petri Net is a graph model for the control behavior of systems)
- Information System SSADM
- Availability of Users the methodology that considers users for their feedback
- Specialized Techniques like knowledge based, graphic based etc...
- HArdware Environment fast speed, restricted computer memory; low level programming can be used
- Safety critical systems independent teams will develop parallel systems
- Imprecise requirements uncertainties or new platform; Prototyping can be used
- 12. Structure vs. Speed of Delivery RAD / JAD or SSADM



Build or buy?





Advantages of off-the-shelf (OTS) software

- Cheaper as supplier can spread development costs over a large number of customers
- Software already exists
 - Can be trialled by potential customer
 - No delay while software being developed
- Where there have been existing users, bugs are likely to have been found and eradicated



Disadvantages of off-the-shelf

- Customer will have same application as everyone else: no competitive advantage, but competitive advantage may come from the way application is used
- Customer may need to change the way they work in order to fit in with OTS application
- Customer does not own the code and cannot change it
- Danger of over-reliance on a single supplier



Software Development Methodologies

Systems development methodology is defined as a standard process followed in an organization to conduct all the steps necessary to analyze, design, implement and maintain information systems

SSADM: Waterfall

https://www.youtube.com/watch?v=t7YMEnnxjJ8

USDP: The **Unified Software Development Process** or **Unified Process** is an iterative and incremental software development process framework. Phases in USDP are:-

- 1. Inception
- 2. Elaboration
- 3. Construction
- 4. Transition https://www.youtube.com/watch?v=Us9eAnCm024



Software Development Methodologies

DSDM (Dynamic System Development

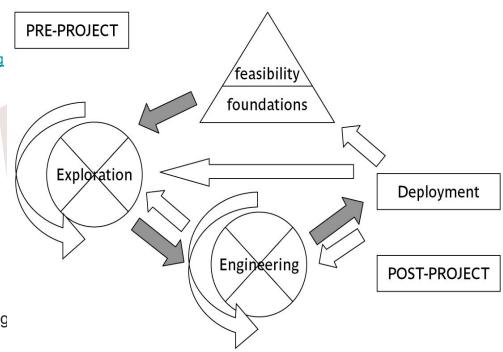
Method) - https://www.youtube.com/watch?v=rllcaPEGpyg

This method provides Four- Phase

Framework:-

- Feasibility / Foundation
- Exploration Cycle
- 3. Engineering Cycle
- 4. Deployment

[DSDM is a **part of Rapid Development Methodology** (RAD) whereas RUP is part of Obeject Oriented Methodolog



https://www.youtube.com/watch?v=jLsbuiB5jeM

www.youtube.com/watch?v=JNHQXpR1hCY

Eight core Atern(2007) /DSDM(1995) principles

- 1. Focus on business need
- 2. Delivery on time use of time-boxing
- 3. Collaborate
- 4. Never compromise quality
- 5. Deliver iteratively
- 6. Build incrementally
- 7. Communicate Continuously
- 8. Demonstrate Control



ATERN'S MoSCoW

Mo - Most important / Must have

S - Should have

Co - Could have

W - Won't have



Genesis of 'Agile' methods

Structured development methods have several disadvantages

- produce large amounts of documentation which can largely remain unread
- documentation has to be kept up to date
- division into specialist groups and need to follow procedures stifles communication
- users can be excluded from decision process
- long lead times to deliver anything etc. etc
- Difficult to accommodate change requests
- Too rigid



The answer? 'Agile' methods?

Agile Methods

- Agile is an umbrella term that refers to a group of development processes:
 - Crystal technologies
 - Atern (formerly DSDM)
 - Feature-driven development
 - Scrum
 - Extreme Programming (XP)
- Similar themes: with Some variations



Pair Programming





Important Themes of Agile Methods

- Base on the incremental approach:
 - At a time, only one increment is planned, developed, and then deployed at the customer site.
- Agile model emphasizes face-to-face communication over written documents.
- An agile project usually includes a customer representative in the team.
- Agile development projects usually deploy Pair programming.
- https://www.youtube.com/watch?v=yagk8a5w_4U
- Case Study: https://www.youtube.com/watch?v=D3iu2kfZ3w4

https://www.youtube.com/watch?v=5fqqJluLoeY



Basic Principles of Agile Methods

- Incremental delivery after each time box
- Face to face communication
- Customer Interaction
- Minimal Documentation
- Pair Programming



Extreme Programming (XP)

It is called Extreme Programming as this approach takes commonsense principles to extreme levels

Four Core values of XP:-

- 1. Communication and Feedback face to face
- 2. Simplicity simplest design that implements user's requirements
- 3. Responsibility developers are the ones who are responsible for the Software Quality
- 4. Courage to throw away the work done earlier, to start fresh

XP refers to increments as RELEASES, each release takes 1-2 months.



Core Practices of XP

- Planning Exercise- features to be added in the next release is negotiated by using ways like <u>Atern's MoSCoW</u>
- 2. Small Releases
- 3. Metaphor using meaningful names for variables, procedures etc.
- 4. Simple Design
- 5. Testing- testing and coding are done at the same time
- 6. Refactoring is that instead of the programmer to keep reusing a bunch of lifeless codes, they can remove them and completely rejuvenate old designs. This way the system is maintained in top shape, and it runs smoothly. The principal goal of refactoring is to improve the quality and save time during the project life cycle.

- 7. Pair Programming
- 8. Collective Ownership
- 9. Continuous Integration (at
- least once a day)
- 10. Forty hours week
- 11. On Site Customers
- 12. Coding Standards

https://www.youtube.com/watch?v=kFM2Vcu-B
Rohttps://www.youtube.com/watch?v=hbFOwq
YIOcU
Deepika Sharma



Project Characteristics that Suggest Suitability of Extreme Programming

- Projects involving new technology or research projects.
 - In this case, the requirements change rapidly and unforeseen technical problems need to be resolved.

Small projects:

 These are easily developed using extreme programming.



SCRUM

- It follows AGILE Principles
- Project is divided into small work parts
- These work parts are developed Incrementally
- Work parts are delivered over time-boxes called SPRINTS
- A Sprint is 2-4 weeks long
- After each Sprint Stakeholders assess the work done

Key Roles & Responsibilities

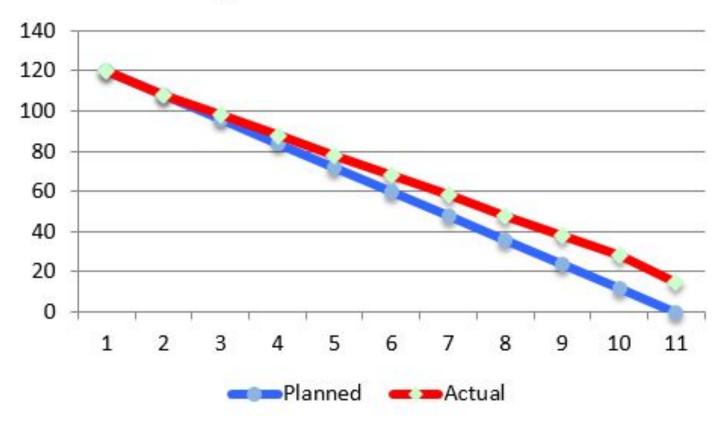
- Product Owner customer
- Scrum Master project manager
- Team Member- Team of five-nine people with cross-functional skill sets.

ARTEFACTS

- Product Backlog this document lists all the features that remain to be developed
- Sprint Backlog initially it contains a list of task taken from the Product Backlog, to be completed in the current Sprint
- Sprint Burndown Chart A chart to show the progress made and work remaining on daily basis



Sprint 2 Burndown





SCRUM CEREMONIES

- 1. SPRINT PLANNING Team works with product owner to plan a Sprint and decide which features are to be included in the upcoming Sprint
- 2. DAILY SCRUM short stand up meeting conducted every morning to review the status of progress, What was done yesterday? Was there any problem faced yesterday? What is to be done today?
- 3. SCRUM REVIEW MEETING at the end of each Sprint new features included in the sprint are shown to the stakeholders to take feedback from them. https://www.youtube.com/watch?v=2Vt7lk8Ublw



Scrum Task Board Template

Company name

Stories	To Do		In Progress	Testing	This is a sample text. Replace it with your own text. This is a sample text. Replace it with your own text.	
This is a sample text. Replace it with your own text.	This is a sample text. Replace it with your own text. This is a sample text. Replace it with your own text. This is a sample text. Replace it with your own text. This is a sample text. Replace it with your own text.		This is a sample text. This is a sample text. This is a sample text.	This is a sample text. This is a sample text. This is a sample text.		
This is a sample text. Replace it with your own text.	This is a sample text. This is a sample text.	This is a sample text. This is a sample text.	This is a sample text. Replace it with your own.	This is a sample text. This is a sample text.	This is a sample text. Replace it with your own text.	



LEAN SOFTWARE DEVELOPMENT

- Lean Process was started in Toyota Production System
- Main theme of this method is to eliminate the causes of DELAYS by removing waste
- Waste is anything that does not add value to customer
- Waste can be
 - Rework (for defect correction)
 - Gold plating
 - Scope creeping
 - Excessive documentation
 - Delay in staffing
- To visualize workflow it uses KANBAN
- KANBAN helps in eliminating the bottlenecks

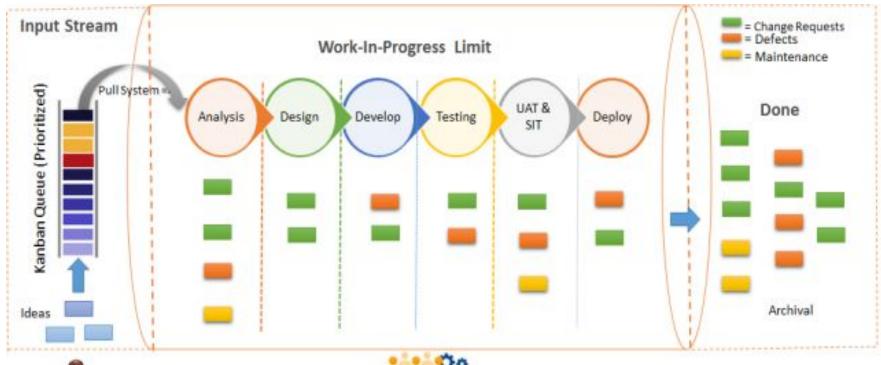


LEAN SOFTWARE DEVELOPMENT

KANBAN

- It gives visibility to WORKFLOW
- KANBAN Board & KANBAN Cards It's like a whiteboard with sticky notes (KANBAN CARD) on it
- On the board Columns are drawn to represent STAGE / Workstations
- CARDS represent the work items waiting to get completed at different development stages.
- CARDS signify, if there is queue up at some stage
- KANBAN enforces WIP(work in progress) limits
- Average time taken by a work item to get completed is called CYCLE TIME









Kanban System

