Project Management and Agile Approaches

System Analysis and Design

School of Software Engineering, Tongji University

Index

- Project Management Fundamentals
- Prototyping
- Agile Approaches
  - Values and Principles
  - SCRUM
  - XP: the Circle of Life
  - Best Practices

Project Management Fundamentals

- Project initiation
- Determining project feasibility
- Activity planning and control
- Project Scheduling
  - Gantt charts
  - PERT diagrams
- Managing systems analysis team members

Project Initiation

- Problems in an Organization
- Defining the Problems
- Set up goals

To Identify Problems	Look for These Specific Signs:
Check output against performance criteria.	• Too many errors • Work completed slowly • Work done incorrectly • Work done incompletely • Work not done at all
Observe behavior of employees.	• High absenteeism • High job dissatisfaction • High job turnover
Listen to external feedback from: Vendors and service providers Customers. Suppliers.	• Complaints • Suggestions for improvement • Loss of sales • Lower sales

Catherine's Catering

A Problem definition example

Problem Definition

Catherine's Catering is experiencing problems with handling the number of routine calls with customers, as well as coordinating with external partners such as suppliers and meeting facilities. The growth in the number of part-time staff is leading to scheduling conflicts and understaffed events.

Issues	Weight
1. Customer contact takes an inordinate amount of time for routine questions.	10
2. Managing part-time employees is time consuming and leads to scheduling errors.	9
3. It is difficult to accommodate last-minute changes for events.	7
4. Supplies are ordered for each event. Often shipments are received several times a day.	6
5. There are often problems communicating changes to event facilities.	5
6. There is little historical information about customers and meals.	3

Objectives

1. Provide a Web system for customers to obtain pricing information and place orders.
2. Create or purchase a human resources system with a scheduling component.
3. After customers have signed an event contract, provide them with Web access to their account and a means for them to update the number of guests. Notify management of changes.
4. Provide a means to determine overall quantities of supplies for events occurring within a concurrent time frame.
5. Provide a system for communicating changes to key personnel at event facilities.
6. Store all event data and make summary information available in a variety of formats.

Requirements

1. The system must be secure.
2. Feedback must be entered by event managers at the close of each event.
3. There must be a means for event facilities to change their contact person.
4. The system must be easy to use by nontechnical people.

Constraints

1. Development costs must not exceed $50,000.
2. The initial website for customer orders must be ready by March 1 to accommodate requests for graduation parties and weddings.

Selection of Projects

- Backing from management
- Appropriate timing of project commitment
- Possibility of improving attainment of organizational goals
- Practical in terms of resources for the system analyst and organization
- Worthwhile project compared with other ways the organization could invest resources

Determining Feasibility

The Three Key Elements of Feasibility

Technical Feasibility
- Add on to present system
- Technology available to meet users' needs

Economic Feasibility
- Systems analysts' time
- Cost of systems study
- Cost of employees' time for study
- Estimated cost of hardware
- Cost of packaged software or software development

Operational Feasibility
- Whether the system will operate when put in service
- Whether the system will be used

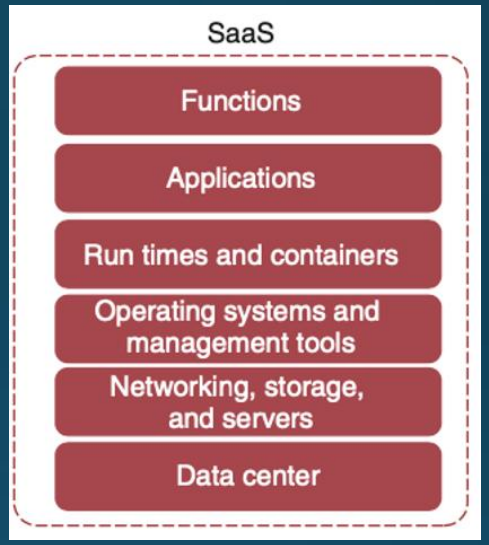
Estimating Workloads

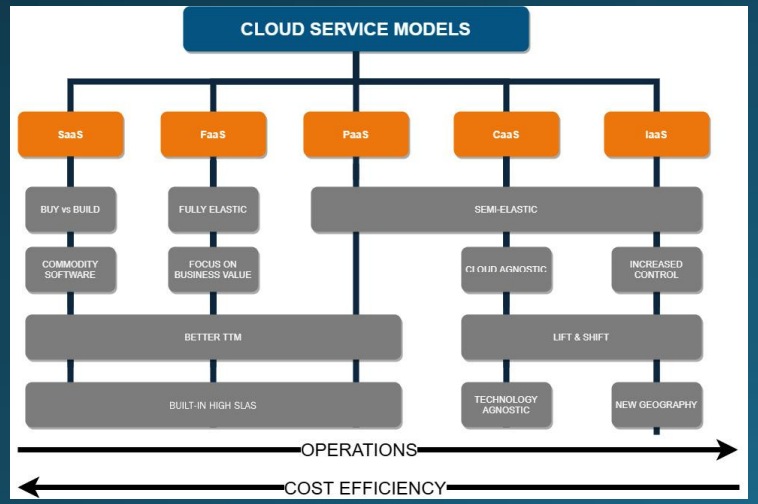
	Existing System	Proposed System
Task	Compare performance of distribution warehouses by running the summary program.	Compare performance of distribution warehouses on the Web-based dashboard.
Method	Computer programs are run when needed; processing is done from the workstation.	Updates occur immediately; processing is done online.
Personnel	Distribution manager	Distribution manager
When and how	Daily: Enter shipments on Excel spreadsheet; verify accuracy of spreadsheet manually; and then write files to backup media. Monthly: Run program that summarizes daily records and prints report; get report and make evaluations.	Daily: Enter shipments on the Web-based system using drop-down boxes. Data are automatically backed up to remote location. Monthly: Compare warehouses online using the performance dashboard; print only if needed.
Human time requirements	Daily: 20 minutes Monthly: 30 minutes	Daily: 10 minutes Monthly: 10 minutes
Computer time requirements	Daily: 20 minutes Monthly: 30 minutes	Daily: 10 minutes Monthly: 10 minutes

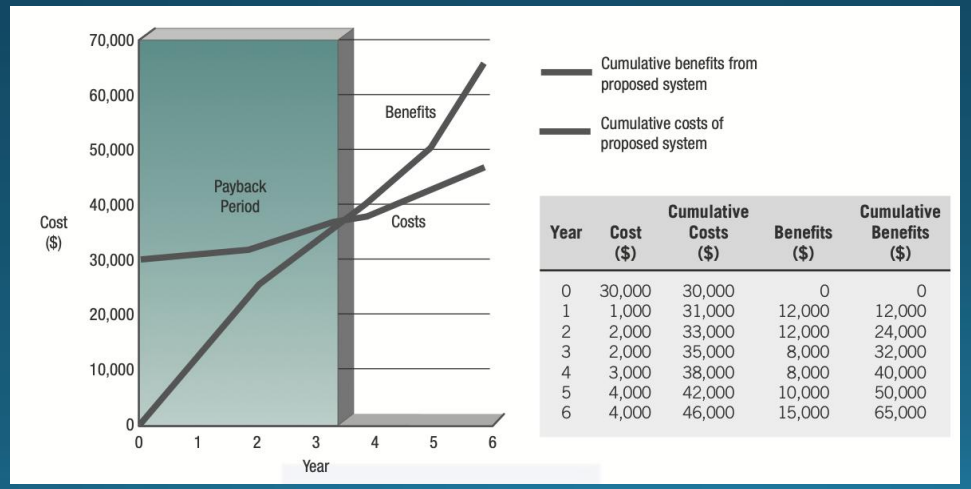
Ascertaining Hardware and Software Needs

- Steps used to determine hardware and software needs:
  - Inventory computer hardware currently available
  - Estimate current and future system workloads
  - Evaluate available hardware and software
  - Choose the vendor
  - Acquire the computer equipment
- Three main categories of cloud computing
  - Software as a Service (SaaS)
  - Infrastructure as a Service (IaaS)
  - Platform as a Service (PaaS)

TTM: Time to Market

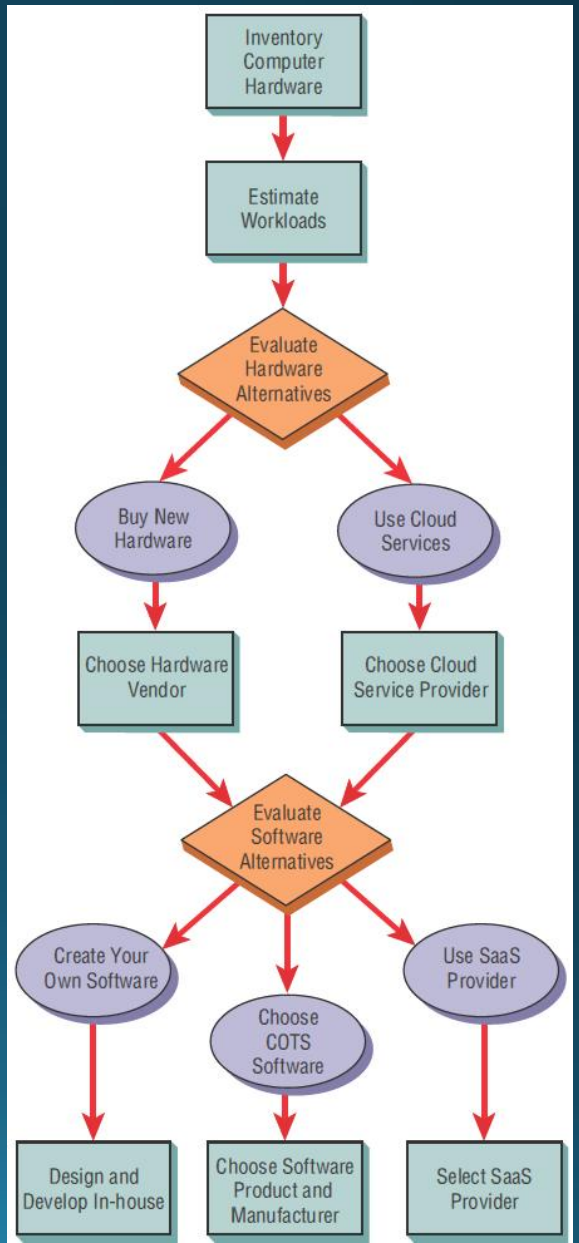






Year	Cost ($)	Cumulative Costs ($)	Benefits ($)	Cumulative Benefits ($)
0	30,000	30,000	0	0
1	1,000	31,000	12,000	12,000
2	2,000	33,000	12,000	24,000
3	2,000	35,000	8,000	32,000
4	3,000	38,000	8,000	40,000
5	4,000	42,000	10,000	50,000
6	4,000	46,000	15,000	65,000

Payback Analysis



Managing Time and Activities

- Often a project needs to be broken down into smaller tasks or activities
- These tasks together make up a work breakdown structure (WBS)
  - Each task or activity contains one deliverable, or tangible outcome, from the activity
  - Each task can be assigned to a single individual or a single group
  - Each task has a responsible person monitoring and controlling performance
- Developing a WBS
  - Decomposition, starting with large ideas, then breaking them down into manageable activities
  - Product oriented, building a website can be broken down into many parts
  - Process-oriented, emphasizes the importance of each phase
- Time Estimation Techniques
  - Relying on experience
  - Using analogies
  - Using three-point estimation
  - Identifying function points
  - Using time estimation software

The Program Evaluation and Review Technique (PERT)

$$\underset{\textit{Beta Distribution}}{PERT} = \frac{O + 4M + P}{6}$$

The Program Evaluation and Review Technique (PERT)

Optimistic (O), Most Likely (M), and Pessimistic (P)

Project Scheduling

Phase	Activity
Analysis	Data gathering Data flow and decision analysis Proposal preparation
Design	Data entry design Input design Output design Data organization
Implementation	Implementation Evaluation

Break apart the major activities into smaller ones.

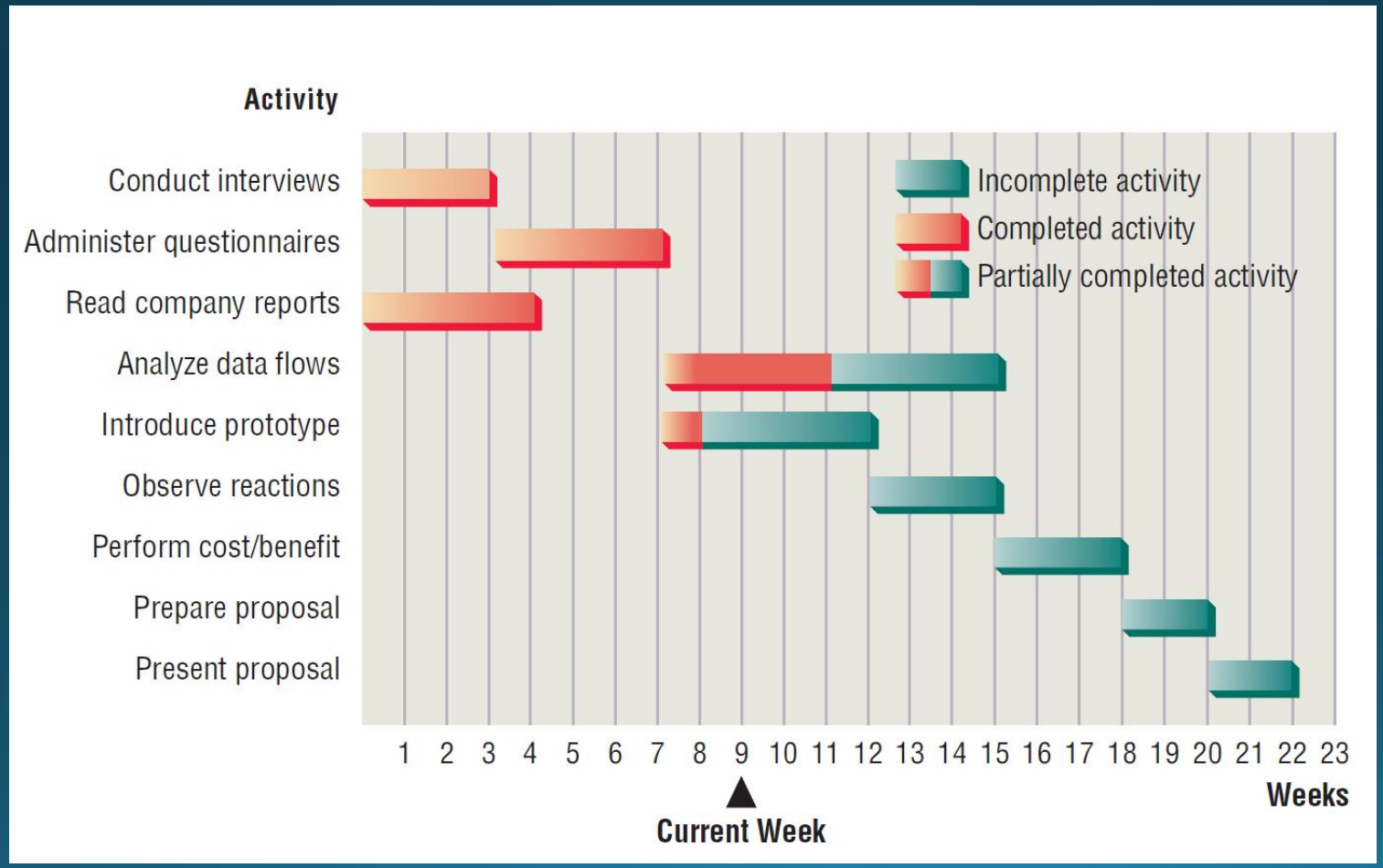
Activity	Detailed Activity	Weeks Required
Data gathering	Conduct interviews	3
	Administer questionnaires	4
	Read company reports	4
	Introduce prototype	5
	Observe reactions to prototype	3
Data flow and decision analysis	Analyze data flow	8
Proposal preparation	Perform cost-benefit analysis	3
	Prepare proposal	2
	Present proposal	2

Break these down further,

then estimate time required.

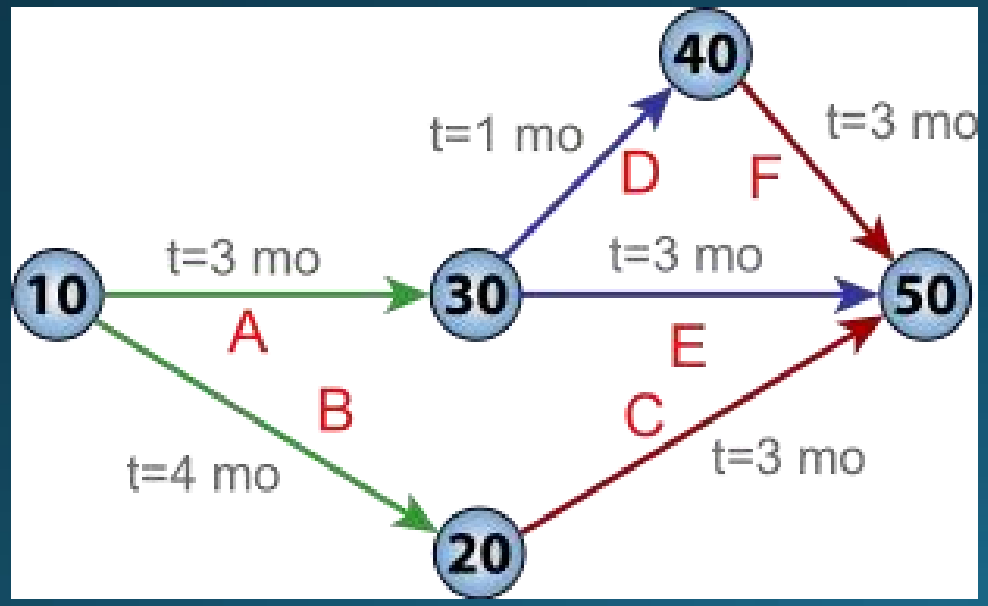
Using Gantt Charts for Project Scheduling

A Gantt chart is a type of bar chart that illustrates a project schedule. Modern Gantt charts also show the dependency relationships between activities and the current schedule status.

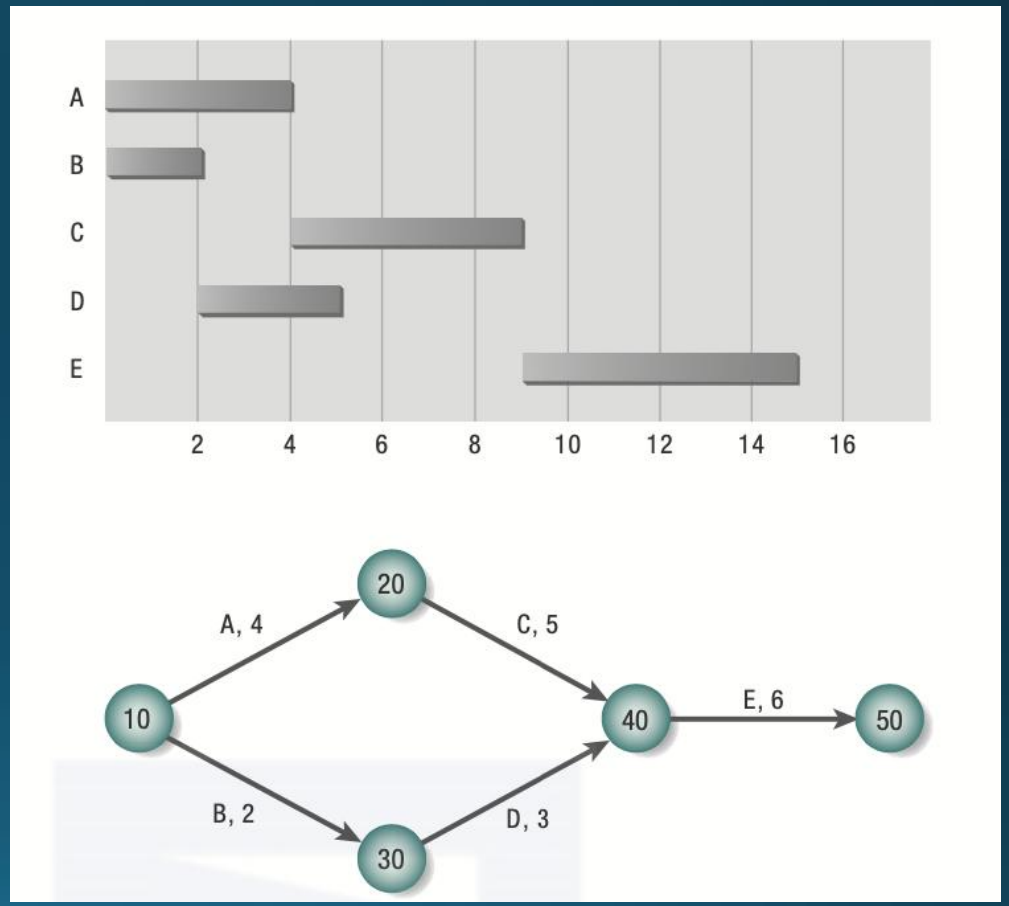


Using PERT

- PERT is an acronym for Program Evaluation and Review Technique.
  - A program is represented by a network of nodes and arrows that are evaluated to determine the critical activities, improve the schedule if necessary, and review progress once the project is undertaken.



PERT network chart for a seven-month project with five milestones (10 through 50) and six activities (A through F).



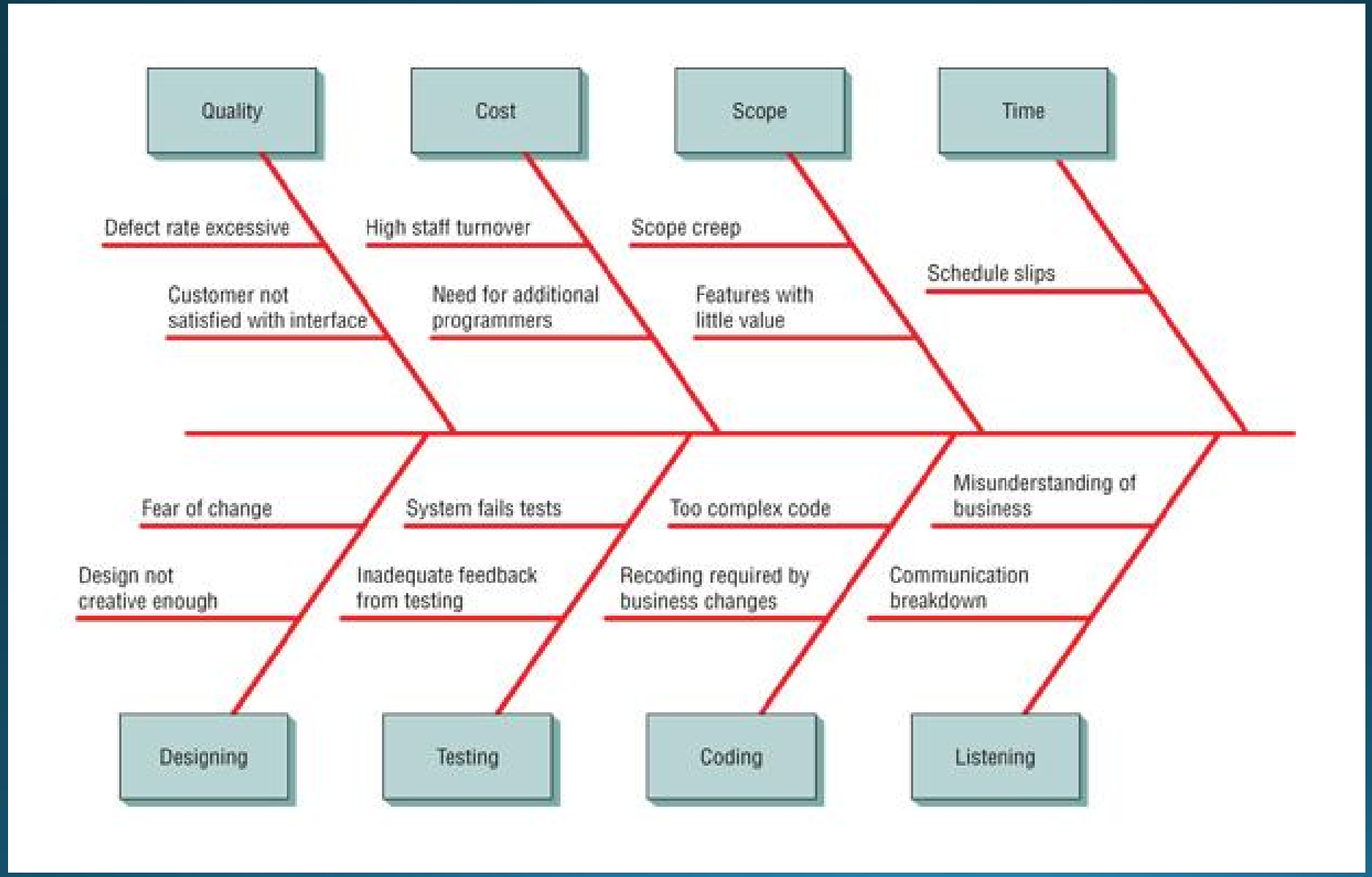
A GANT chart compared with a PERT diagram

https://en.wikipedia.org/wiki/Program_evaluation_and_review_technique

Controlling a Project

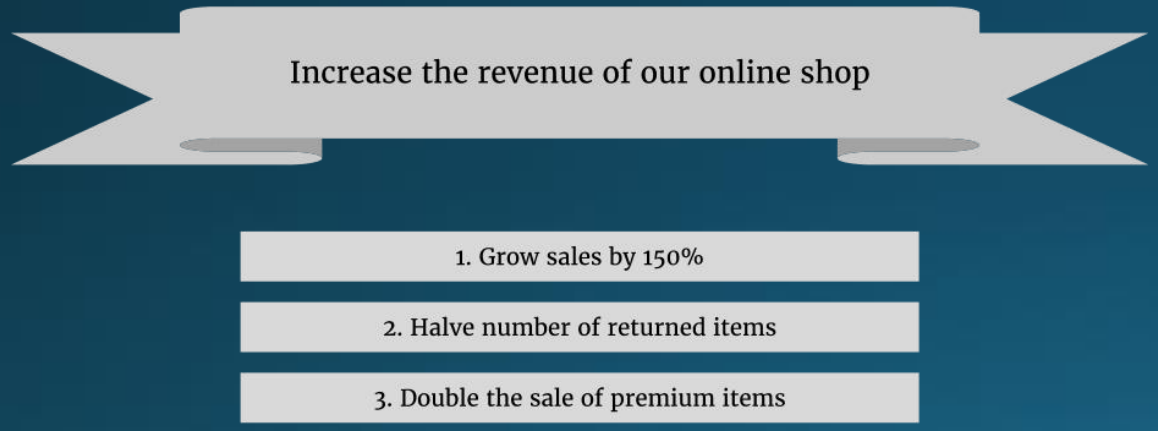
- Estimating Costs and Preparing the Budget
  - Basing estimates on similar projects, also called the top-down approach
  - Building bottom-up estimates
  - Using parametric modeling
- Managing Risk
  - Project failures may be prevented by:
    - Training
    - Experience
    - Learning why other projects have failed
  - Fishbone diagram systematically lists all of the possible problems that can occur

Fishbone Diagram

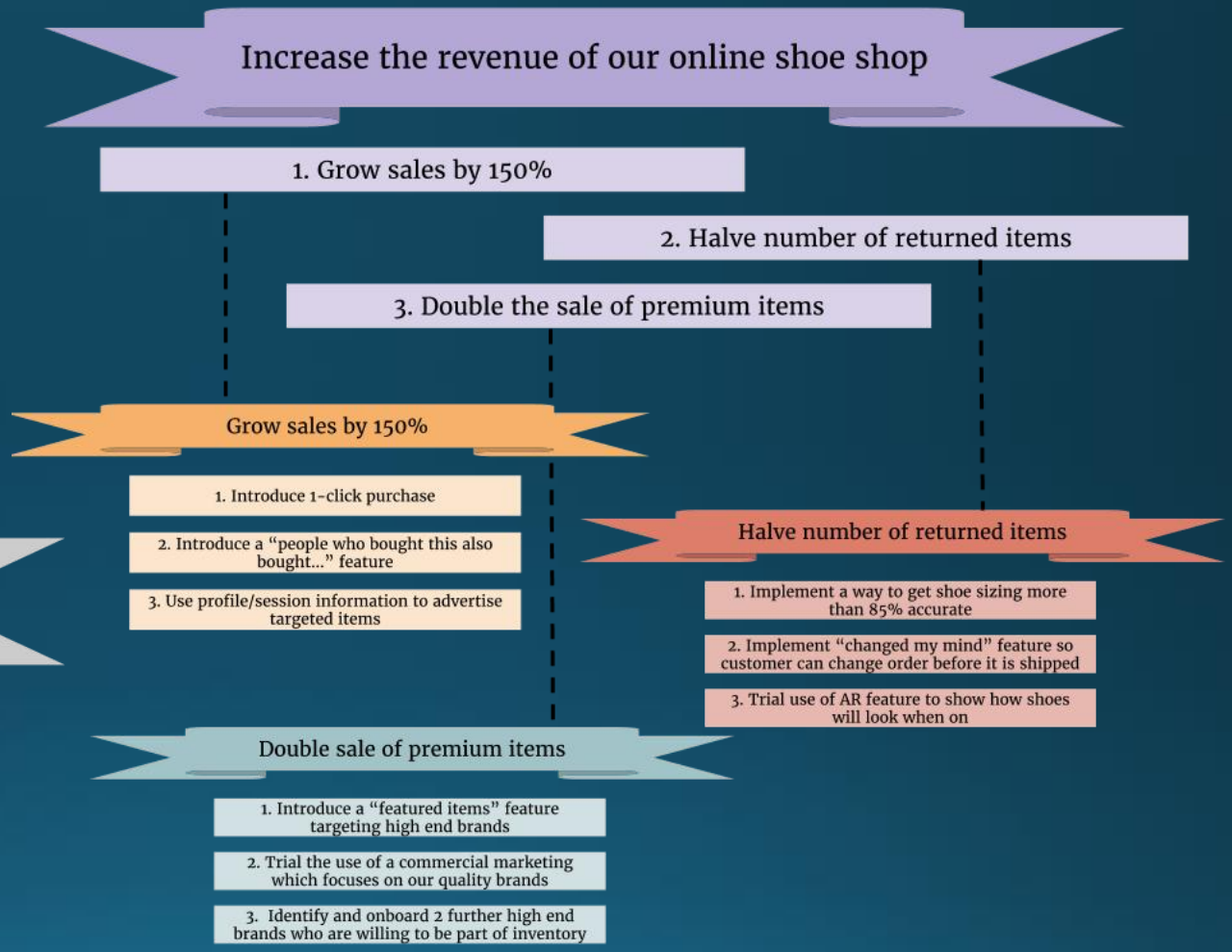


Team Management

- Assembling a team
- Team communication strategies
- Project productivity goals
- Team member motivation



An example OKR for an online shop



From KPI (Key Performance Indicator) to OKR (Objectives and Key Results)

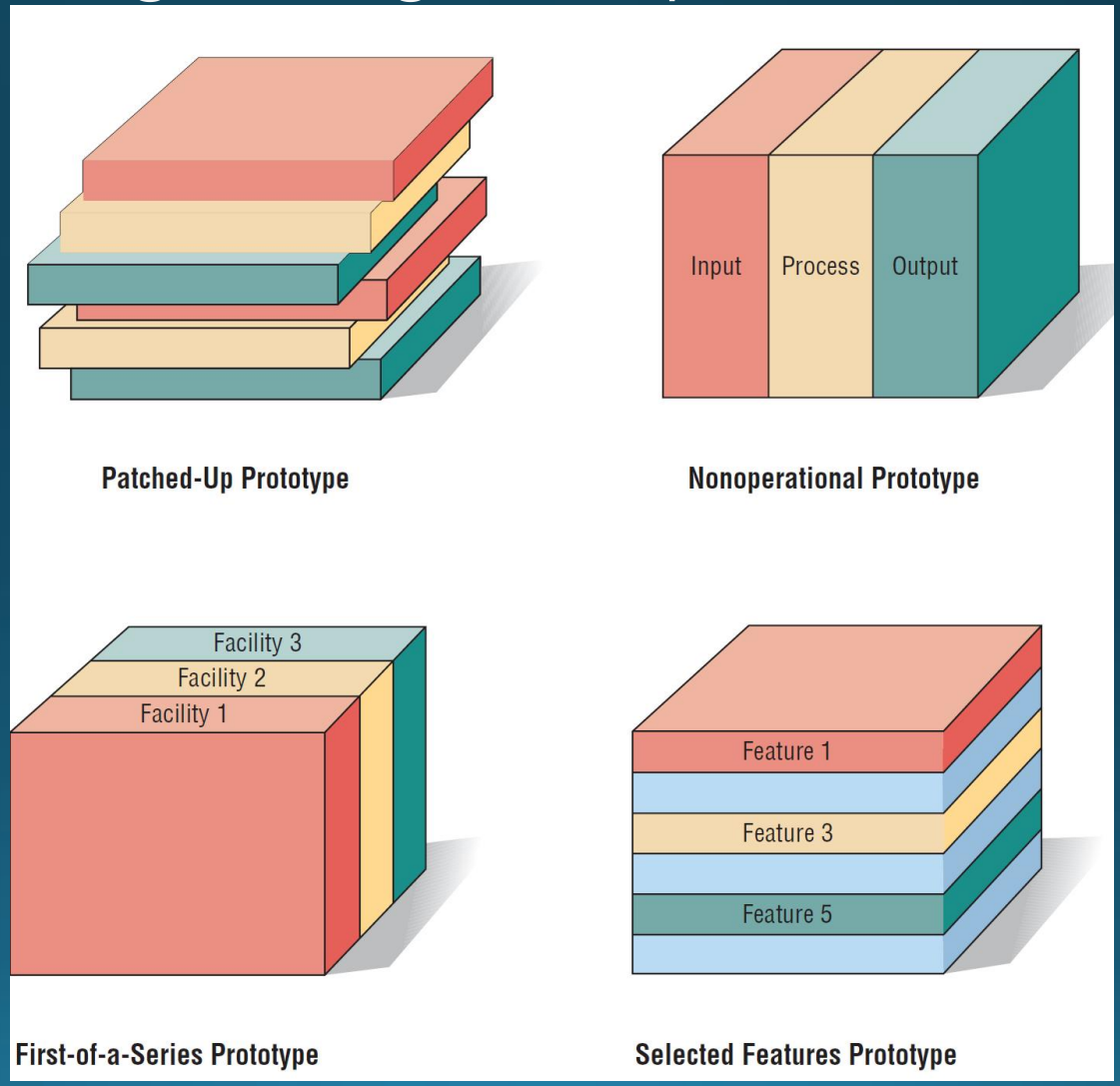
OKRs were first introduced by Andy Grove at Intel Corp when he was CEO. They were then taken to Google by Grove's former colleague, John Doerr. Since then, many high-tech companies have used them including Uber, LinkedIn, Netflix, and so on.

The System Proposal

- Cover letter
- Title page of project
- Table of contents
- Executive summary
- Outline of systems study with appropriate documentation
- Detailed results of the systems study
- Systems alternatives
- Systems analysts recommendations
- Summary
- Appendices

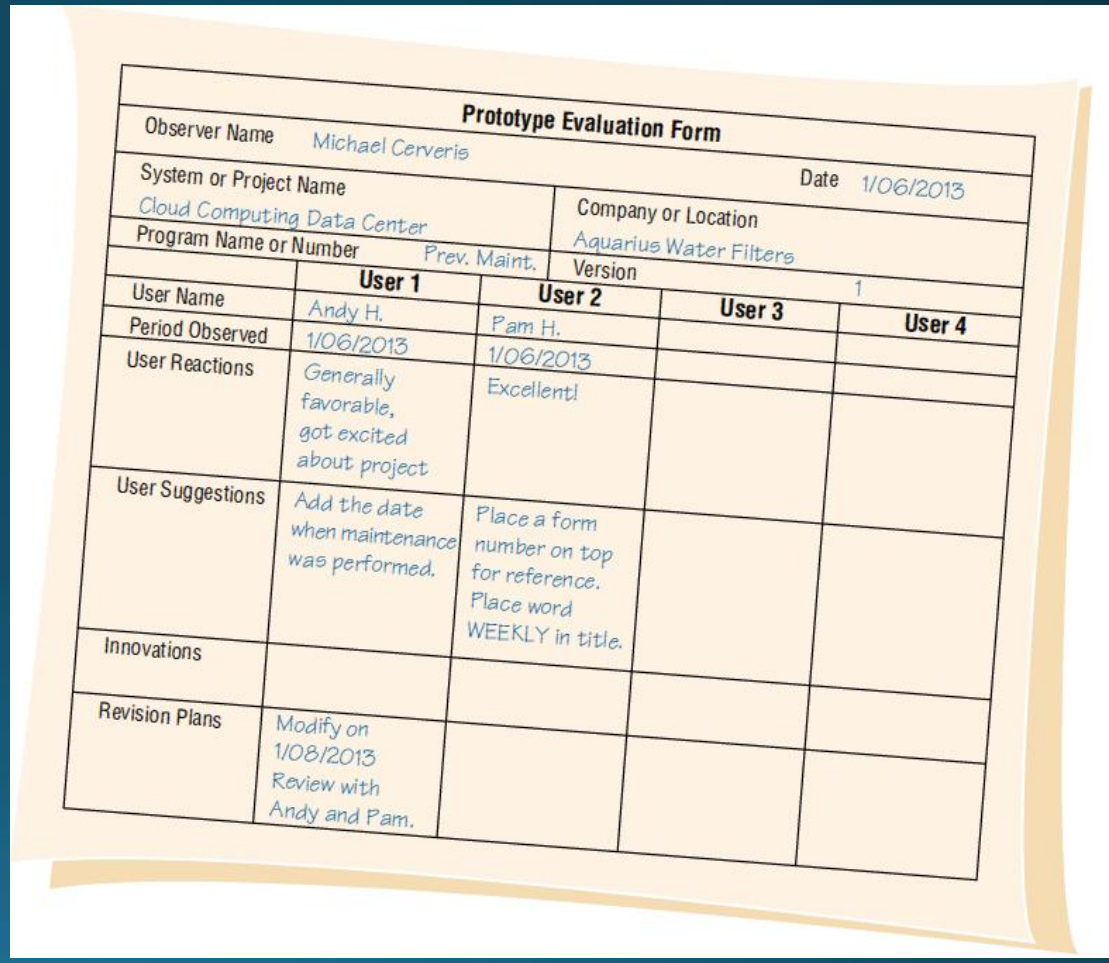
Prototyping

- Prototyping is an information-gathering technique useful in seeking
  - User reactions
  - Suggestions
  - Innovations
  - Revision plans
- Four Kinds of Prototypes



Guidelines for Developing a Prototype

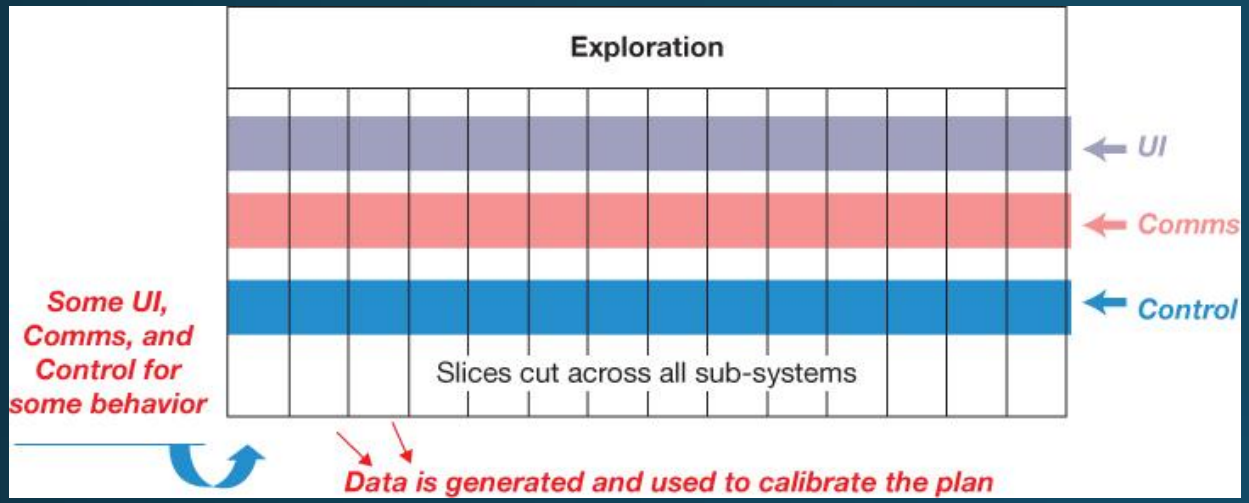
- Work in manageable modules
- Build the prototype rapidly
- Modify the prototype in successive iterations
- Stress the user interface



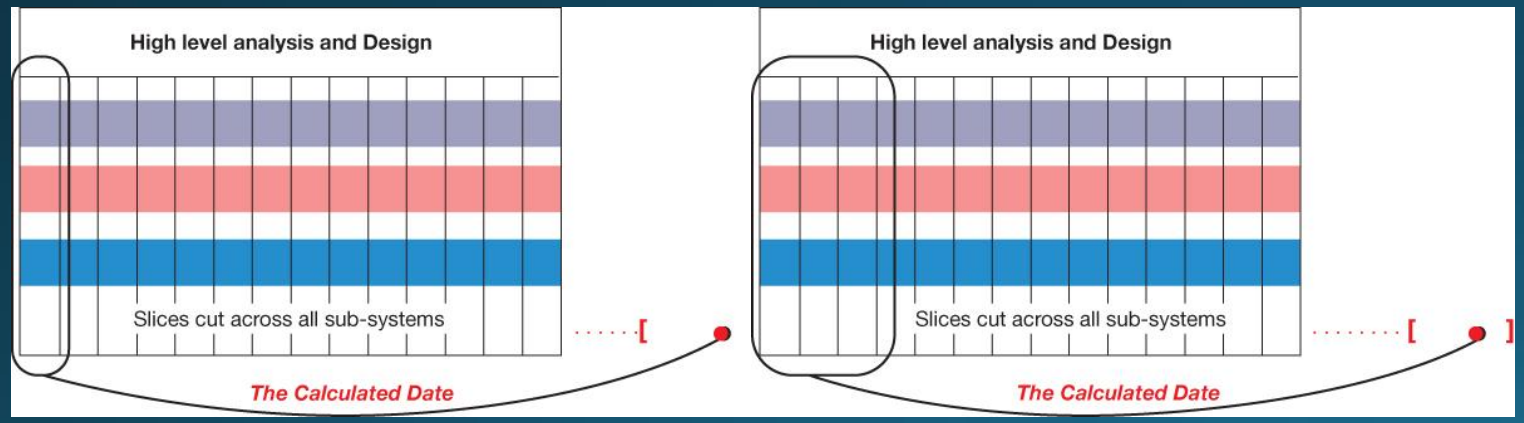
Prototype Evaluation Form

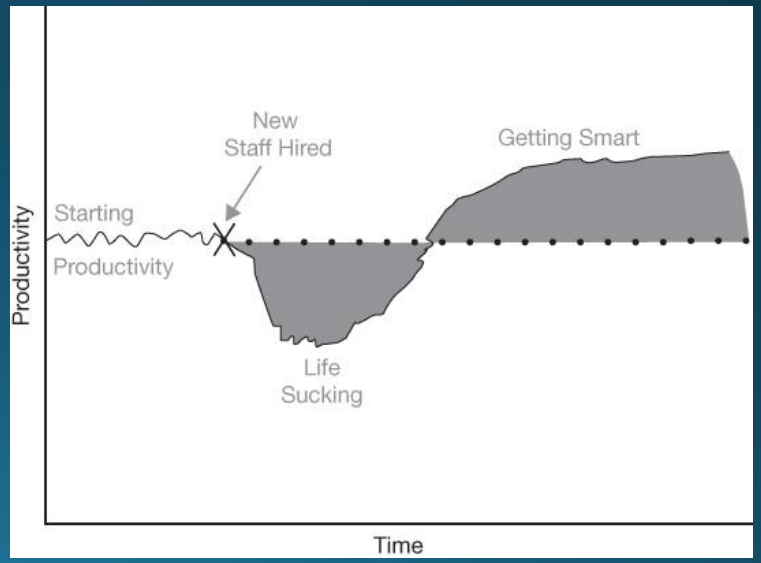
Observer Name: Michael Cerveris			Date: 1/06/2013	
System or Project Name: Cloud Computing Data Center		Company or Location: Aquarius Water Filters		
Program Name or Number: Prev. Maint.		Version: 1		
	User 1	**User 2**	**User 3**	**User 4**
User Name	Andy H.	Pam H.		
Period Observed	1/06/2013	1/06/2013		
User Reactions	Generally favorable, got excited about project	Excellent!		
User Suggestions	Add the date when maintenance was performed.	Place a form number on top for reference. Place word WEEKLY in title.		
Innovations				
Revision Plans	Modify on 1/08/2013 Review with Andy and Pam.			

The Agile Approach



- Agile is a process wherein a project is subdivided into iterations.
- The output of each iteration is measured and used to continuously evaluate the schedule.
- Features are implemented in the order of business value so that the most valuable things are implemented first.
- Quality is kept as high as possible.
- The schedule is primarily managed by manipulating scope.





Agile Development is adaptive rather than predictive; people-oriented rather than process-oriented.

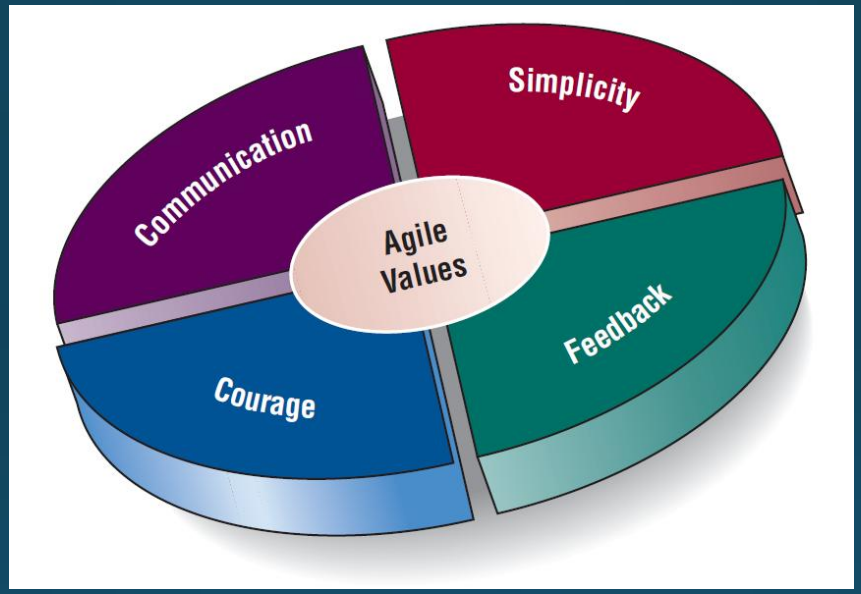
Martin Fowler

The actual effect of adding more members to the team

Adding manpower to a late project makes it later.

Values and Principles of Agile Modeling

- Agile methods are a collection of innovative, user-centered approaches to systems development



- Four Basic Activities of Agile Modeling
  - Coding
  - Testing
  - Listening
  - Designing

The Basic Principles of Agile Modeling

- **Satisfy the customer** through delivery of working software
- **Embrace change**, even if introduced late in development
- Continue to deliver functioning software **incrementally and frequently**
- Encourage customers and analysts to **work together** daily
- Trust motivated individuals to get the job done
- Promote face-to-face conversation
- Concentrate on **getting software to work**
- **Encourage continuous, regular, and sustainable development**
- Adopt agility with attention to **mindful design**
- Support self-organizing teams
- Provide rapid feedback
- **Encourage quality**
- Review and adjust behavior occasionally
- **Adopt simplicity**

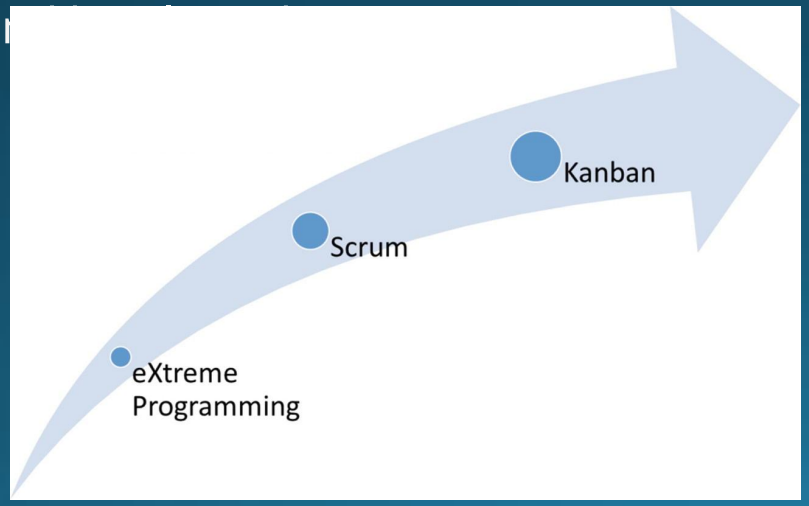
Four Resource Control Variables of Agile Modeling

- Time
- Cost
- Quality
  - Acceptance tests are a collaborative effort between business analysts, QA, and the developers.
  - Business analysts specify the happy paths.
  - QA's role is to write the unhappy paths.
  - Developers work with QA and business analysts to ensure that the tests make sense from a technical point of view.
- Scope

The four-quadrant game

Agile Methodologies

- There are three most common methodologies/methods to implement Agile principles: eXtreme programming, Scrum, and Kanban.
  - XP is the most prescriptive (specific) of the Agile methodologies, for which the main focus is **prescription of engineering** processes.
  - Scrum is one of the most popular, if not *the* most popular, of Agile methodologies. Similar to XP, it has short releases. Those iterations are called Sprints.
  - Kanban was invented by Toyota in the 1940s to reduce idle time during manufacturing. In the software world, it is basically executing work as it comes and using a board with notes to track the progress a

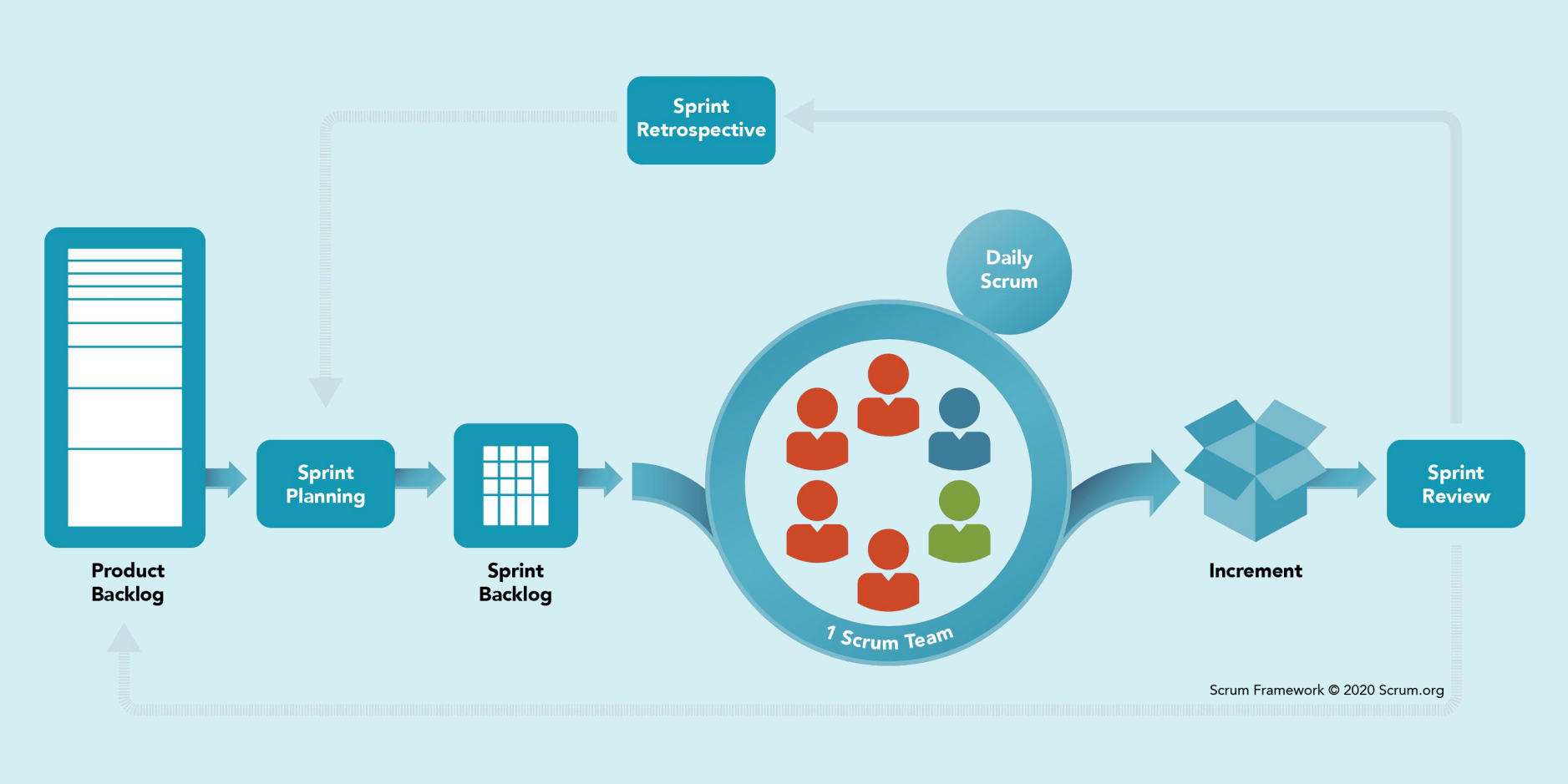


Moving from prescriptive to more adaptive

Scrum

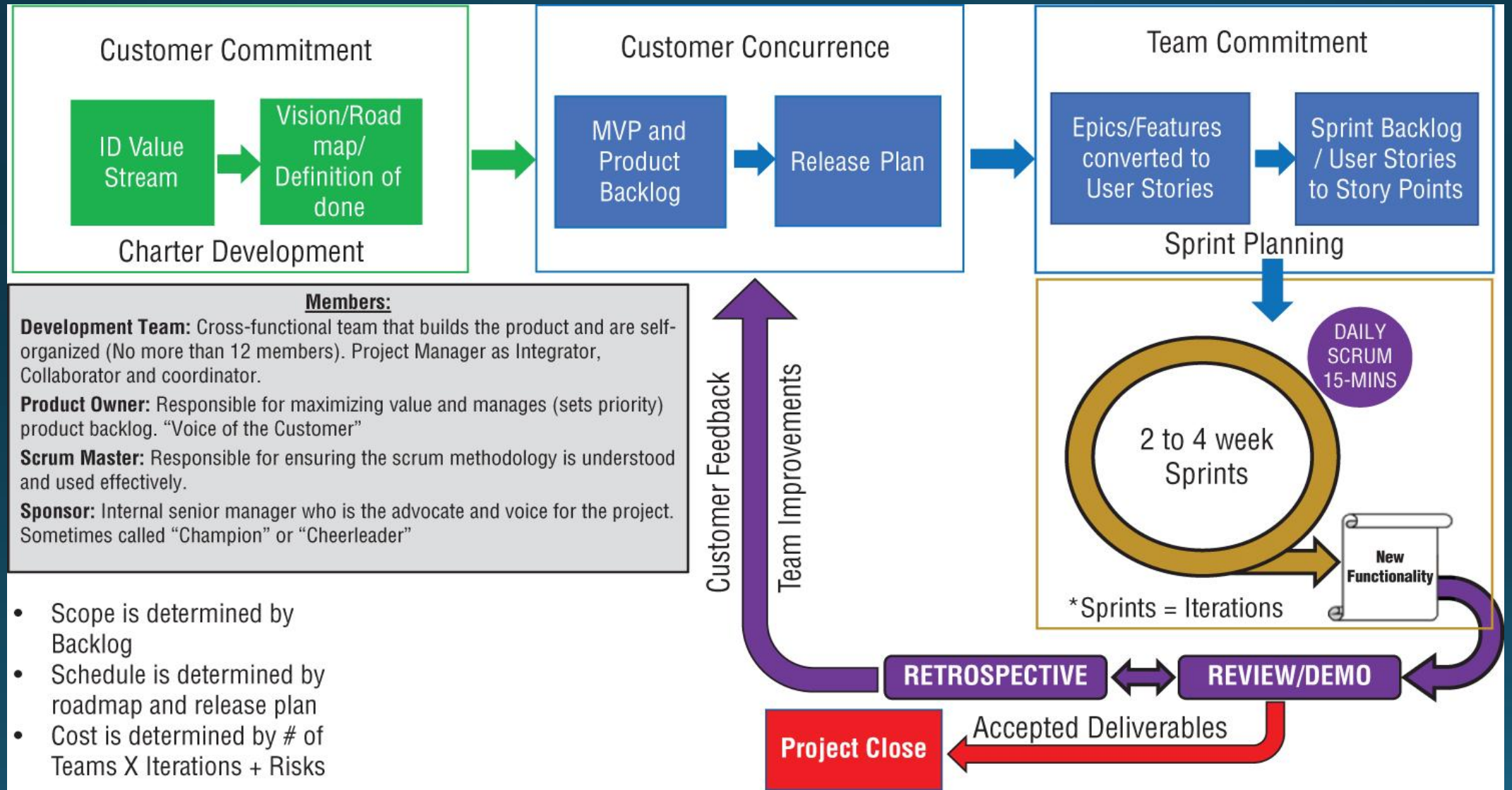
- Scrum is a framework for developing and sustaining complex products.
- In a nutshell, Scrum requires a Scrum Master to foster an environment where:
  - A Product Owner orders the work for a complex problem into a Product Backlog.
  - The Scrum Team turns a selection of the work into an Increment of value during a Sprint.
  - The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.
  - Repeat

Scrum in Action



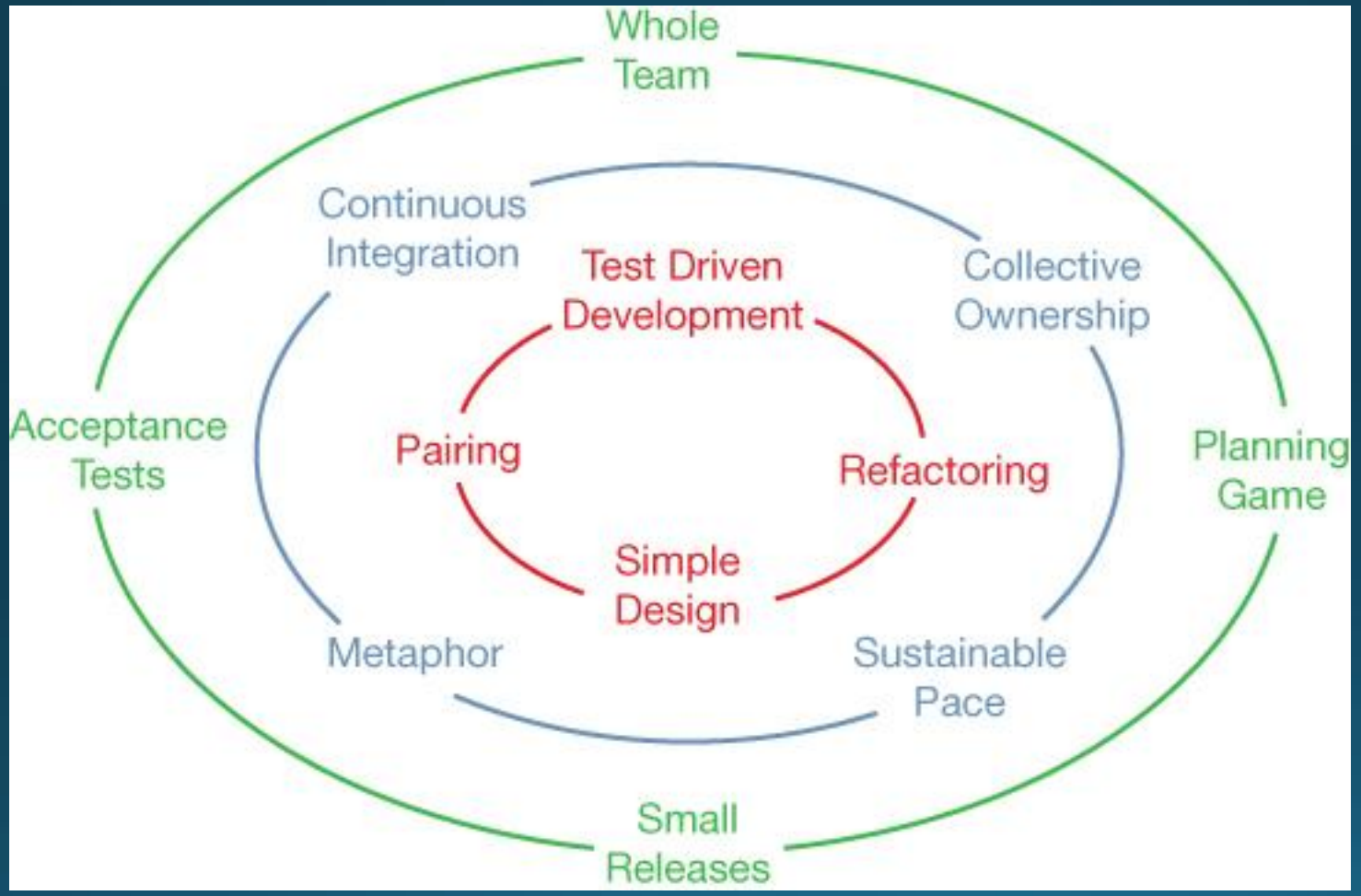
http://jeffsutherland.org/scrum/scrum_plop.pdf
https://www.scrumguides.org/
https://www.scrum.org/

Example:
Personalized Fitness and Nutrition Mobile App



- Scope is determined by Backlog
- Schedule is determined by roadmap and release plan
- Cost is determined by # of Teams X Iterations + Risks

The practice of XP: *the Circle of Life*



The outer ring shows the business-facing practices of XP.
This ring is essentially the equivalent of the Scrum process.

The Business-facing Practice

- The ***Planning Game*** tells us how to break down a project into features, stories, and tasks. It provides guidance for the estimation, prioritization, and scheduling of those features, stories, and tasks.
- ***Small Releases*** guides the team to work in bite-sized chunks.
- ***Acceptance Tests*** provide the definition of "done" for features, stories, and tasks. It shows the team how to lay down unambiguous completion criteria.
- ***Whole Team*** conveys the notion that a software development team is composed of many different functions, including programmers, testers, and managers, who all work together toward the same goal.

The Team-facing Practice

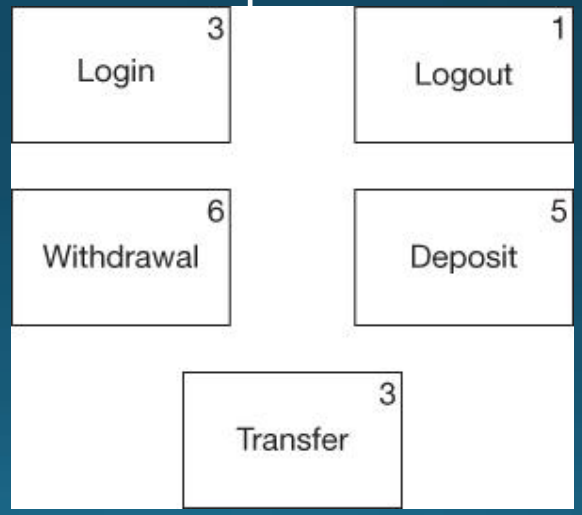
- ***Sustainable Pace*** is the practice that keeps a development team from burning their resources too quickly and running out of steam before the finish line.
- ***Collective Ownership*** ensures that the team does not divide the project into a set of knowledge silos.
- ***Continuous Integration*** keeps the team focused on closing the feedback loop frequently enough to know where they are at all times.
- ***Metaphor*** is the practice that creates and promulgates the vocabulary and language that the team and the business use to communicate about the system.

The Technical-facing Practices

- ***Pairing*** is the practice that keeps the technical team sharing knowledge, reviewing, and collaborating at a level that drives innovation and accuracy.
- ***Simple Design*** is the practice that guides the team to prevent wasted effort.
- ***Refactoring*** encourages continual improvement and refinement of all work products.
- ***Test Driven Development*** is the safety line that the technical team uses to go quickly while maintaining the highest quality.

Business Practices

- Planning: Estimates should be as accurate as possible, but only as precise as necessary to keep the cost of estimation low.
- Trivariate Analysis: *best-case*, *nominal-case*, and *worst-case*
- Stories and Points
  - Example: The Login story gets assigned three points
  - INVEST rules (*details in next page*)



- Planning Iteration one
  - The iteration begins with the *Iteration Planning Meeting* (IPM). This meeting should be scheduled to be one-twentieth the duration of the iteration.

Stories in Agile Planning

User stories are simple statements that we use as reminders of features. We try not to record too much detail when we write the story because we know that those details will likely change.

Stories follow a simple set of guidelines that we remember with the acronym *INVEST*.

- **I: Independent.** User stories are independent of each other.
- **N: Negotiable.** This is another reason why we don't write down all the details.
- **V: Valuable.** The story must have clear and quantifiable value to the business.
- **E: Estimable.** A user story must be concrete enough to allow the developers to estimate it.
- **S: Small.** A user story should not be larger than one or two developers can implement in a single iteration.
- **T: Testable.** The business should be able to articulate tests that will prove that the story has been completed.

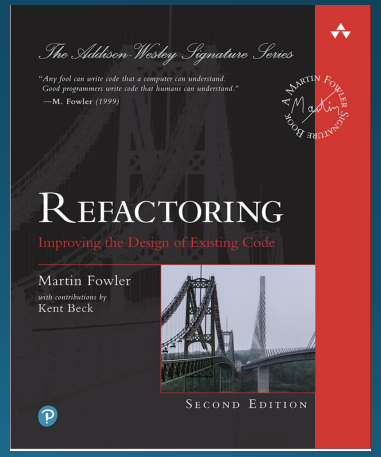
Team Practices

- Metaphor
  - The idea is that in order for the team to communicate effectively, they require a constrained and disciplined vocabulary of terms and concepts. Example: https://en.wikipedia.org/wiki/Chrysler_Comprehensive_Compensation_System
- Domain-Driven Design
  - What the team needs is a model of the problem domain, which is described by a vocabulary that everyone agrees on.
- Sustainable Pace
- Collective Ownership
  - No one owns the code in an Agile project. The code is owned by the team as a whole. Any member of the team can check out and improve any module in the project at any time. The team owns the code *collectively*.
- Continuous Integration
- Stand-up Meeting (Optional, ~ 10 minutes)
  - What did I do since the last meeting?
  - What will I do until the next meeting?
  - What is in my way?

Technical Practices

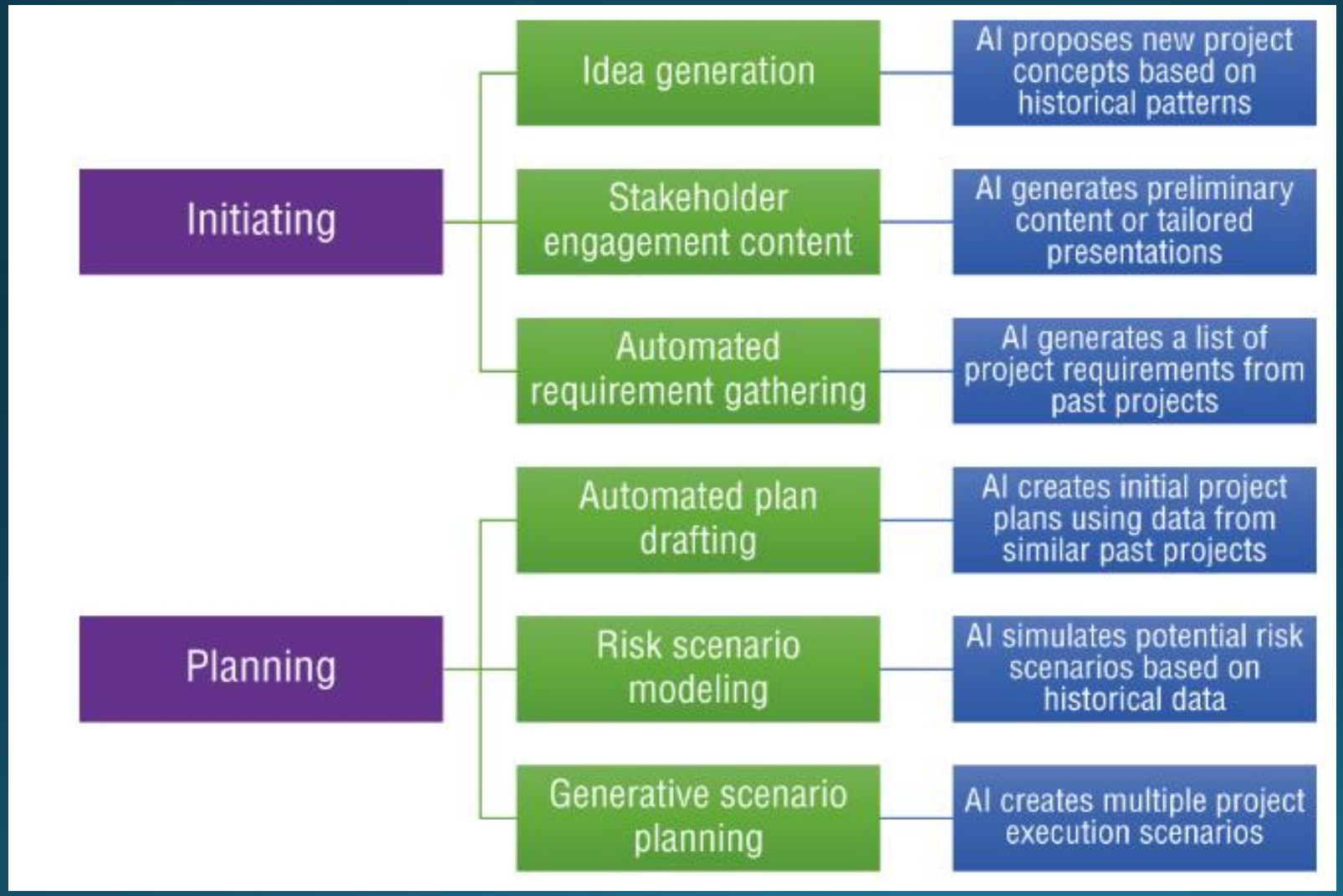
- Test Driven Development
  - Do not write any production code until you have first written a test that fails due to the lack of that code.
  - Do not write more of a test than is sufficient to fail—and failing to compile counts as a failure.
  - Do not write more production code than is sufficient to pass the currently failing test.
- Refactoring
  - Refactoring is the practice of *improving* the structure of the code without altering the behavior, as defined by the tests.
  - Make changes to the names, the classes, the functions, and the expressions without breaking any of the tests.
  - Improve the structure of the system, without affecting the behavior.

  Fowler, M. 2019. *Refactoring: Improving the Design of Existing Code*, 2nd ed. Boston, MA: Addison-Wesley.

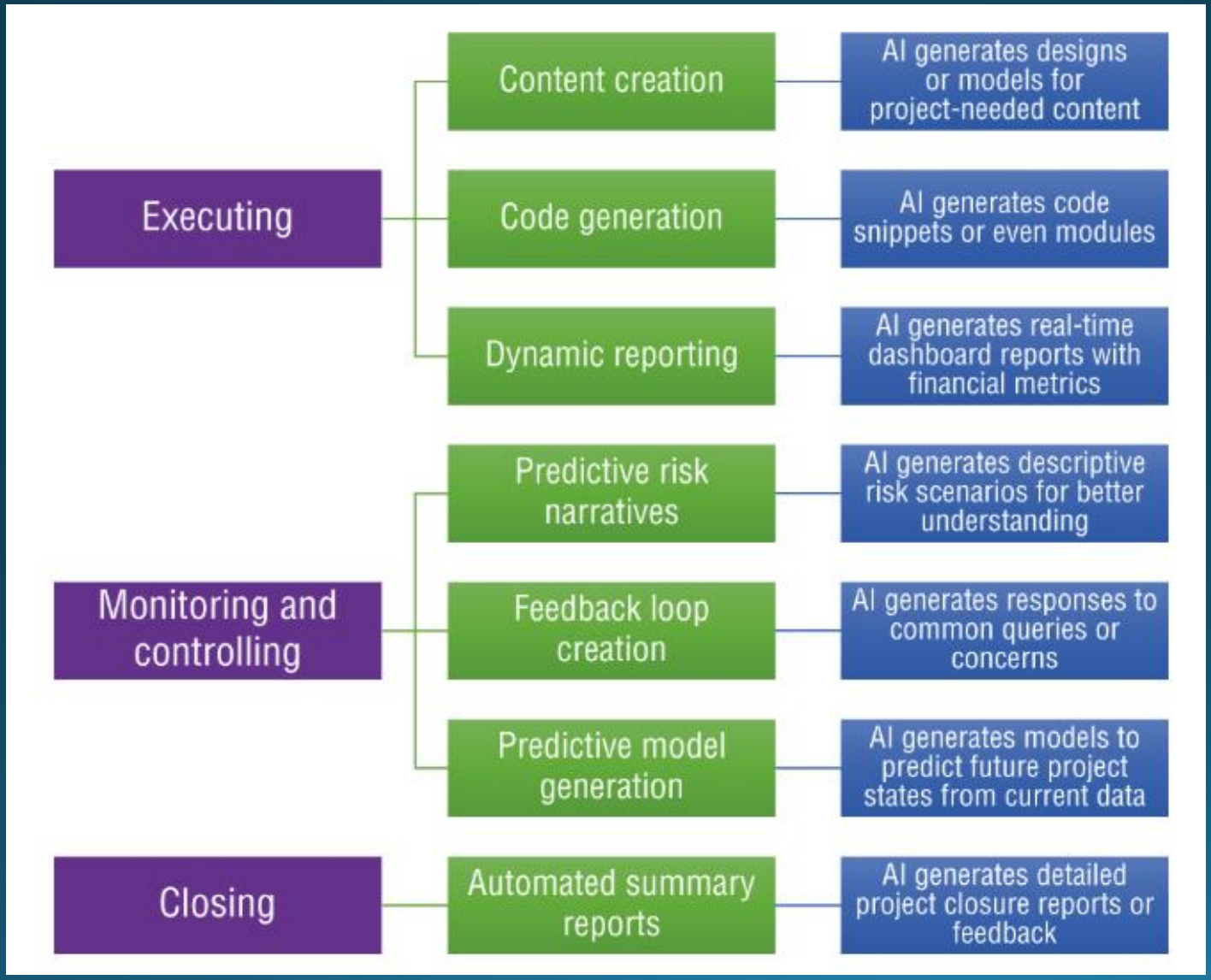


- Simple Design
  - Pass all the tests.
  - Reveal the intent.
  - Remove duplication.
  - Decrease elements.
- Pairing Programming
  - Pairing as Code Review
  - A naive calculation would suggest that a team that pairs 50% of the time would suffer something less than 8% in productivity.

Discussion: AI-Driven Project Management (1)



Discussion: AI-Driven Project Management (2)



Further Reading

- Manifesto for Agile Software Development
  - http://agilemanifesto.org/
- The 2020 Scrum Guide
  - https://www.scrumguides.org/scrum-guide.html