Systems Analysis and Design More SE

CIS641

Erik Fredericks // frederer@gvsu.edu

Adapted from materials provided by Gregory Schymik and the textbook (Systems Analysis and Design 5th/6th Ed.)

Outline

CONTINUATION!
Project management
Requirements

GitHub demo! (hard stop at some point during the lecture to make sure you see this)

Ethical discussion!

https://www.gamedeveloper.com/business/unity-to-start-charging-fee-pegged-to-game-installs

gamedeveloper.com/business/unity-to-start-charging-fee-pegged-to-game-installs

informa

Game Developer | Job Board | Podcast | Articles | Events | Blogs | About Us | STAY UPDATED

Business | 8 MIN READ | NEWS

Unity introducing new fee attached to game installs

Unity's new "Runtime Fee" will kick in after developers cross specific revenue and install thresholds.

Bryant Francis
Senior Editor

September 12, 2023

[redacted]

The Runtime Fee will kick in after developers cross specific revenue and install thresholds that scale with different subscription plans. For those on Unity Personal or Unity Plus licenses, the fee will kick in after a project crosses both $200,000 in revenue over 12 months and 200,000 total installs.

Developers operating on the Unity Pro or Unity Enterprise licenses will be granted a higher threshold before they begin owing fees. For those devs, the charge will kick in after a title earns $1 million over that same 12-month period and passes 1 million installs.

According to a breakdown sent to Game Developer by Unity, fees will be charged on a monthly basis. The amount charged per install will also vary depending on the license.

Unity Personal and Plus developers will pay a flat fee of 20 cents per install. Pro and Enterprise users will pay a smaller per-install fee that scales downward with the number of installations over the initial threshold. Pro user fees will begin at 15 cents per install and scale downward to as low as .02 cents per install, while Enterprise user fees will begin at 12.5 cents per install and scale down to as low as .01 cents per install.

Fees will also vary depending on what region a game is monetizing in. Unity has set an "emerging market monthly rate" for developers monetizing in countries like India. In those regions, Personal and Plus users would owe .02 cents per install, Pro users would owe .01 cents per install, and Enterprise users would owe .005 cents per install.

Developers of free-to-play games (which have a much higher install rate than premium titles) will have the option to offset this fee by adopting other Unity services, such as its LevelPlay advertising mediation service.

An update!

Well, the last one I could reliably find:
https://www.polygon.com/23885373/unity-technologies-install-fee-pricing-change

"Unity Technologies announced Friday that it will no longer force developers into that fee structure for the current version of its Unity software — but said that the **fee will apply to future versions of the engine**, should a developer choose to update. However, **people using Unity Personal won't have to pay at all**. (Developers can use Unity Personal, the free version of the engine, as long as their company's combined funding and revenue stays below $200,000 — an increase from the current $100,000 cap.)

Unity also said that for situations where the Runtime Fee does apply, it will kick in only when a project has met two thresholds: $1 million in gross revenue over the past 12 months, and 1 million initial engagements (which is the fancy, but more precise, term that Unity is using in place of "installs")."

But first...

Let's talk about the research paper I put up on Blackboard!

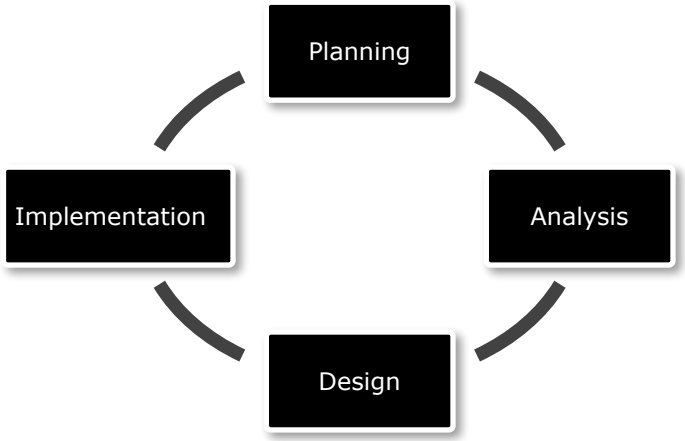
Things we care about:

1) What is the **motivation**?
2) What is the **technique**?
3) What are the **results**?
4) How could we take this work **further**?
5) Is it even **valid** research?

What are your thoughts on the paper?

?

SDLC

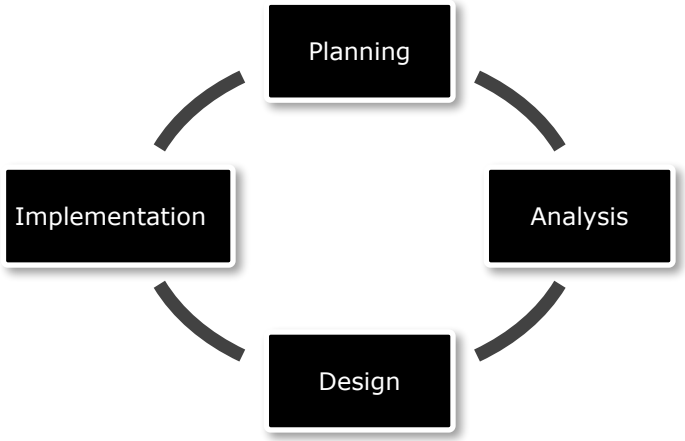


Questions.....

- Planning phase
  - Why should we build this system?
  - What value does it provide?
  - How long will it take to build?

- Analysis phase
  - Who will use it?
  - What should the system do for us?
  - Where & when will it be used?

SDLC

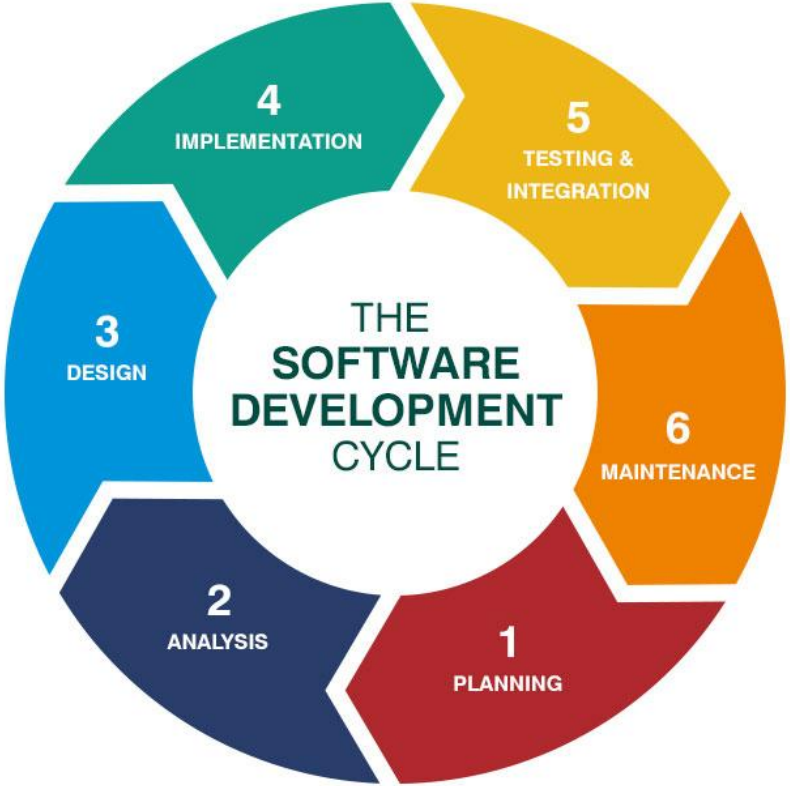


Questions.....

- Design phase
  - How should we build it?

- Implementation Phase
  - How (who?) shall we construct it?
  - How will we test it?
  - How will we deploy it?
  - What is the support plan?

Software Design Lifecycle



Breakout - Small Groups

Read the following case study

(Discuss for 5 minutes and then come back with a group answer)

(Nominate one person to share your thoughts)

Barbara Singleton, manager of western regional sales at the WAMAP Company, requested that the IS department develop a sales force management and tracking system that would enable her to better monitor the performance of her sales staff . Unfortunately, owing to the massive backlog of work facing the IS department, her request was given a low priority. After six months of inaction by the IS department, Barbara decided to take matters into her own hands. Based on the advice of friends, Barbara purchased simple database software and constructed a sales force management and tracking system on her own. Although Barbara's system has been "completed" for about six weeks, it still has many features that do not work correctly, and some functions are full of errors. Barbara's assistant is so mistrustful of the system that she has secretly gone back to using her old paper-based system, because it is much more reliable. Over dinner one evening, Barbara complained to a systems analyst friend, "I don't know what went wrong with this project. It seemed pretty simple to me. **Those IS guys wanted me to follow this elaborate set of steps and tasks, but I didn't think all that really applied to a PC-based system**. I just thought I could build this system and tweak it around until I got what I wanted without all the fuss and bother of the methodology the IS guys were pushing. I mean, doesn't that just apply to their big, expensive systems?"

- **Assuming you are Barbara's systems analyst friend, how would you respond to her complaint?**

Discuss

Ok groups, what should Barbara have done?

Methodologies

Code and Test

- "Debugging the empty source file"

Structured Development

- Waterfall Development
- Parallel Development

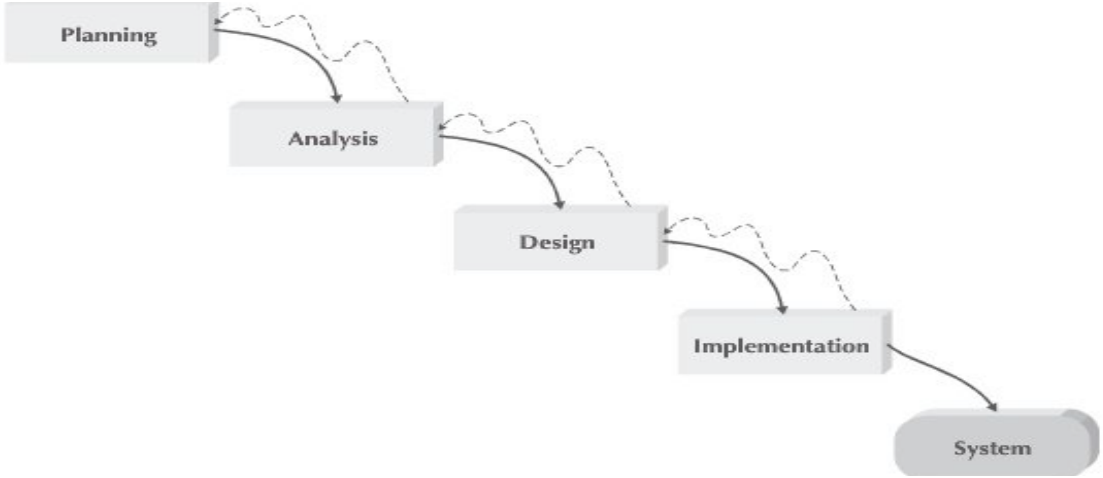
Rapid Application Development

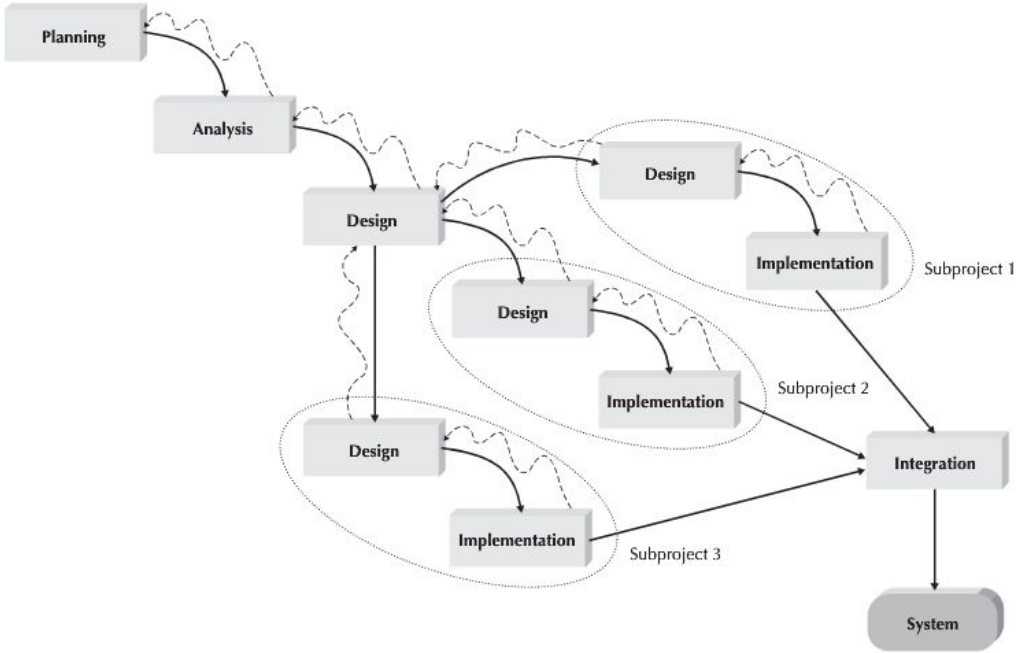
- Phased
- Prototyping

Agile Development

- eXtreme Programming
- SCRUM

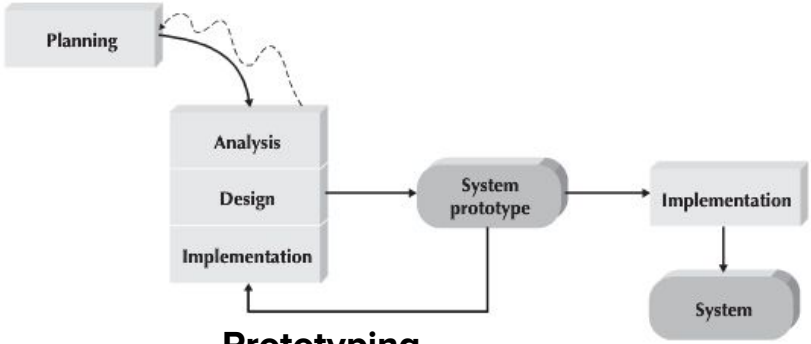
Structured





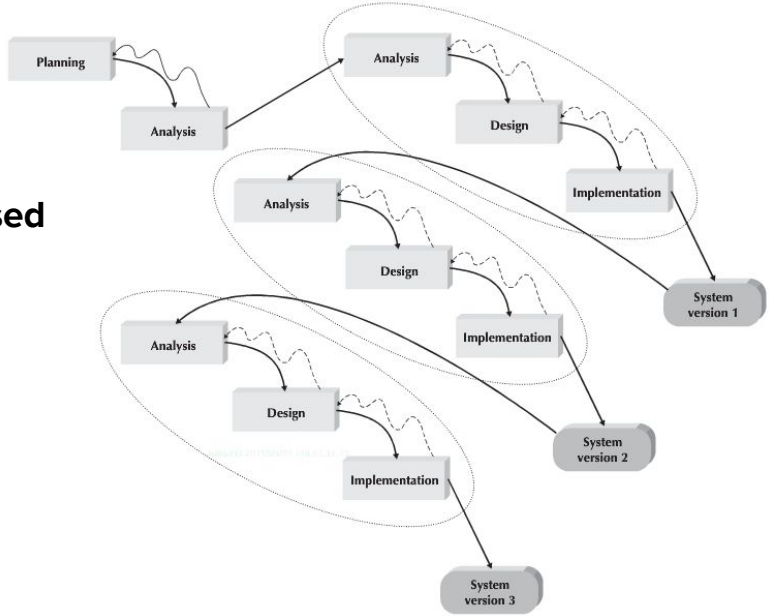
Parallel development tasks ("chunks")

RAD

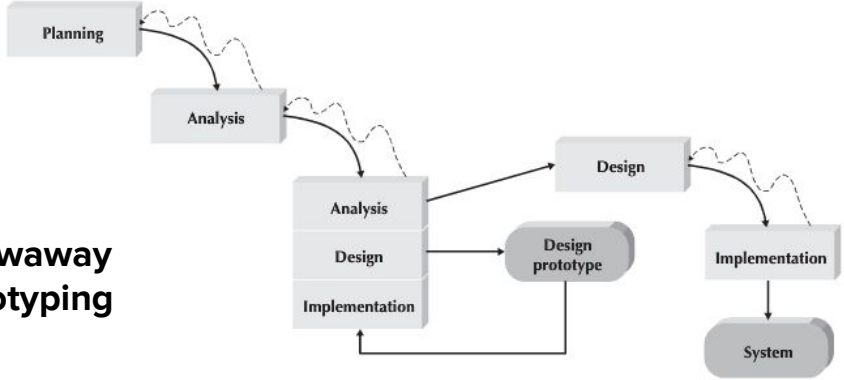


Prototyping

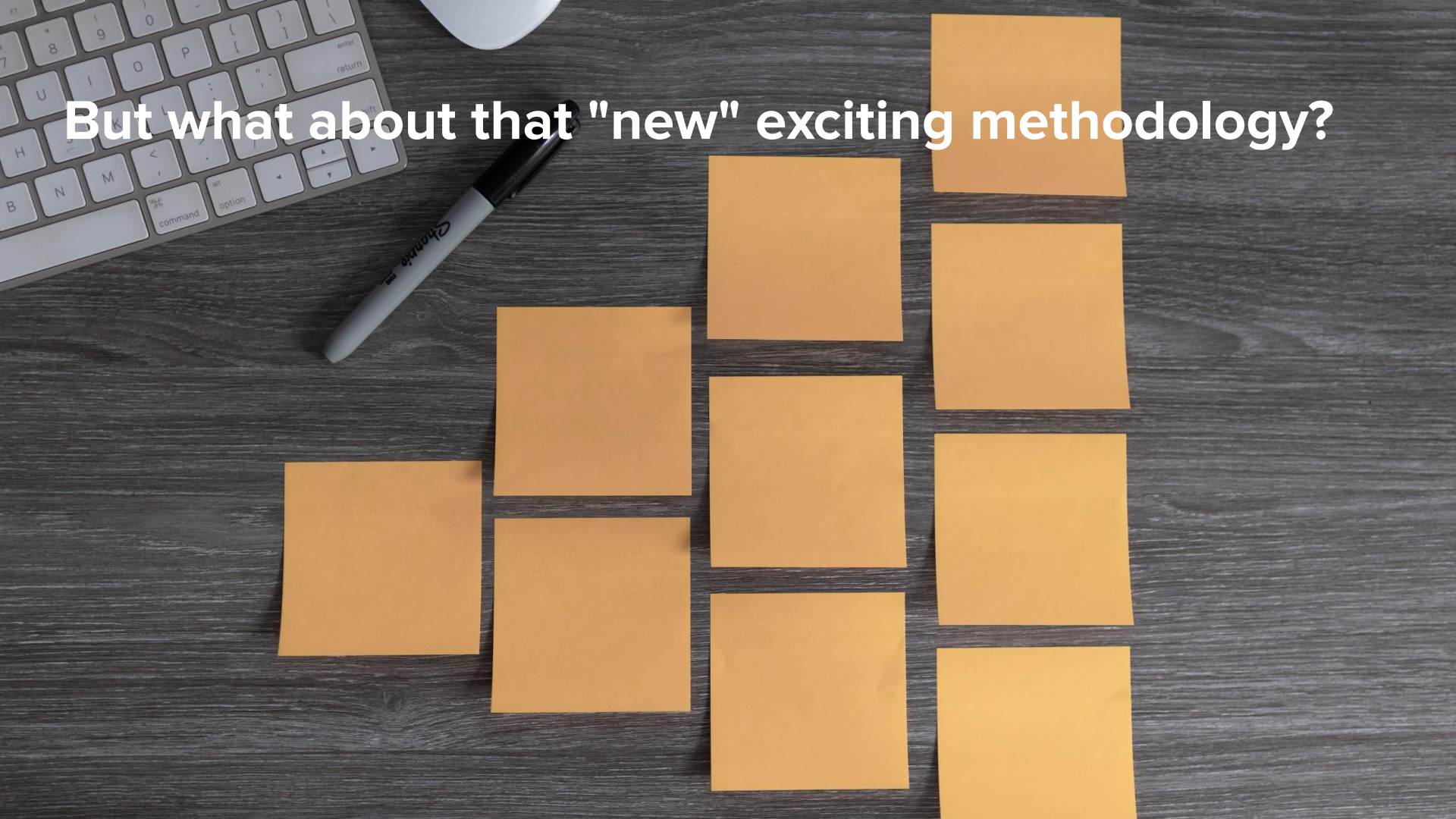
Phased



Throwaway prototyping



But what about that "new" exciting methodology?



Agile

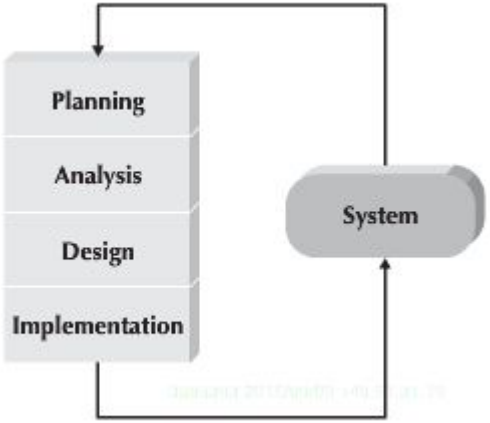


Figure 2-1 The Agile Triangle

- Software is delivered early and continuously through the development process, satisfying the customer.
- Changing requirements are embraced regardless of when they occur in the development process.
- Working software is delivered frequently to the customer.
- Customers and developers work together to solve the business problem.
- Motivated individuals create solutions; provide them the tools and environment they need, and trust them to deliver.
- Face-to-face communication within the development team is the most efficient and effective method of gathering requirements.

Agile

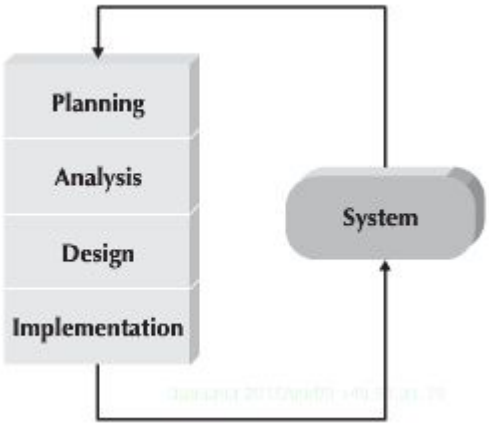
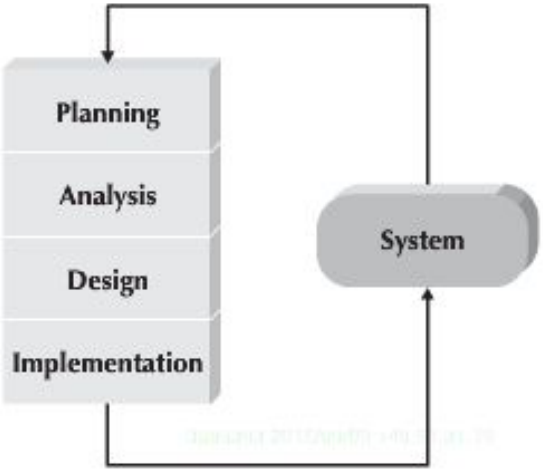


Figure 2-1 The Agile Triangle

- The primary measure of progress is working, executing software.
- Both customers and developers should work at a pace that is sustainable. That is, the level of work could be maintained indefinitely without any worker burnout.
- Agility is heightened through attention to both technical excellence and good design
- Simplicity, the avoidance of unnecessary work, is essential.
- Self-organizing teams develop the best architectures, requirements, and designs.
- Development teams regularly reflect on how to improve their development processes.

Is Agile that simple? We just start writing code?



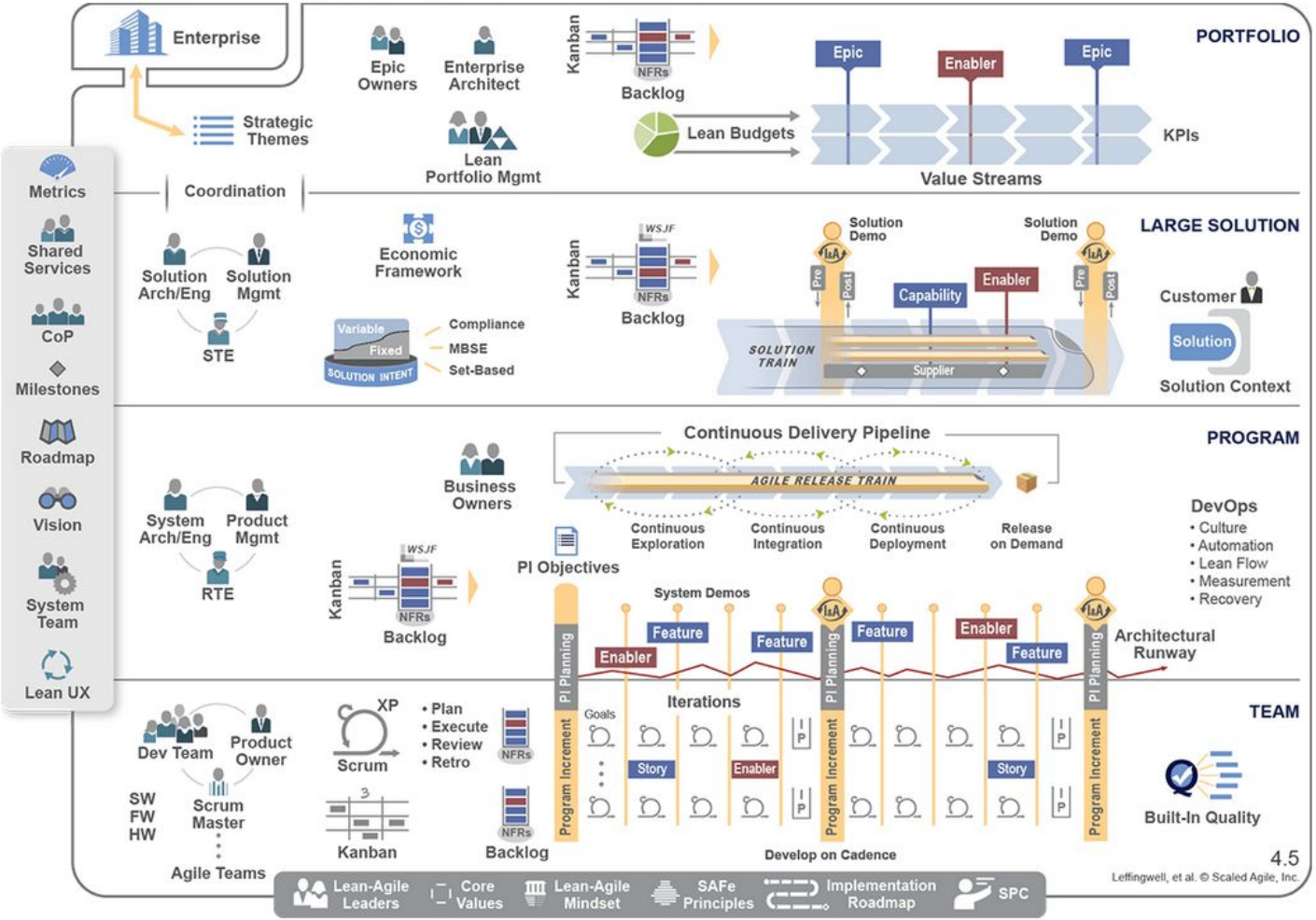
N° 2

GitHub Demo

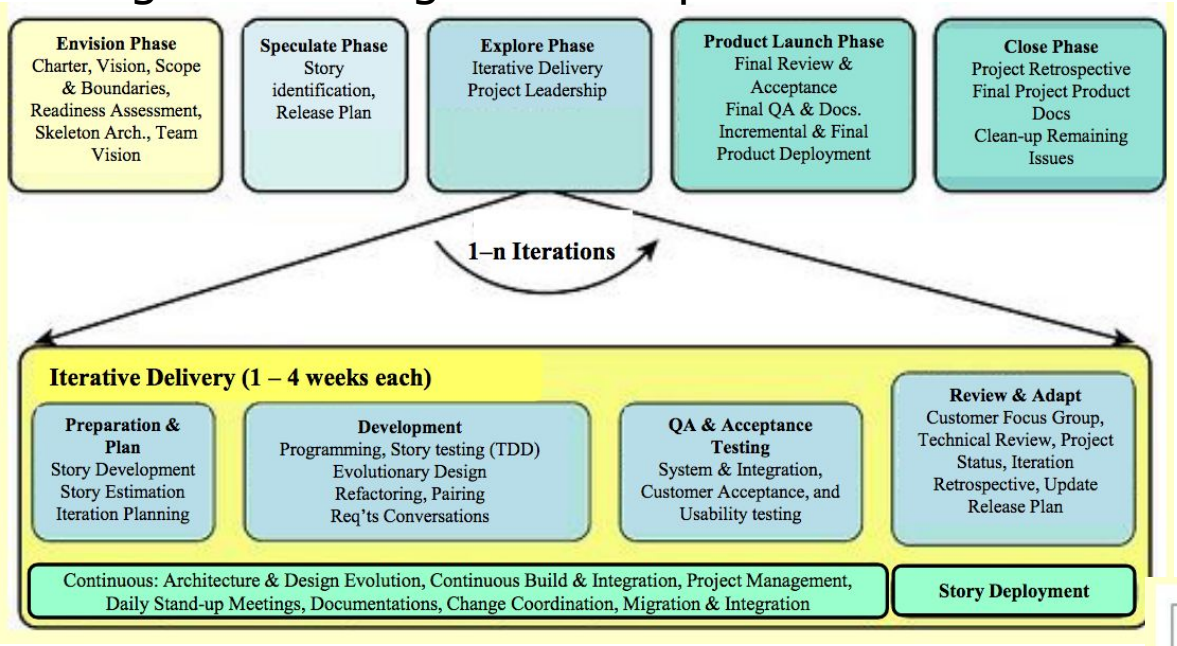
```
$ git pull
$ <make changes>
$ git add .
$ git commit -m '<your message'>
$ git push
```

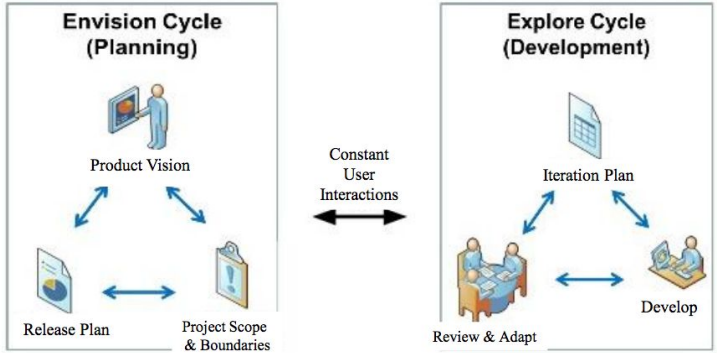
1) New repo
2) Steps above
3) GitHub Pages

SAFe® for Lean Enterprises — Full SAFe



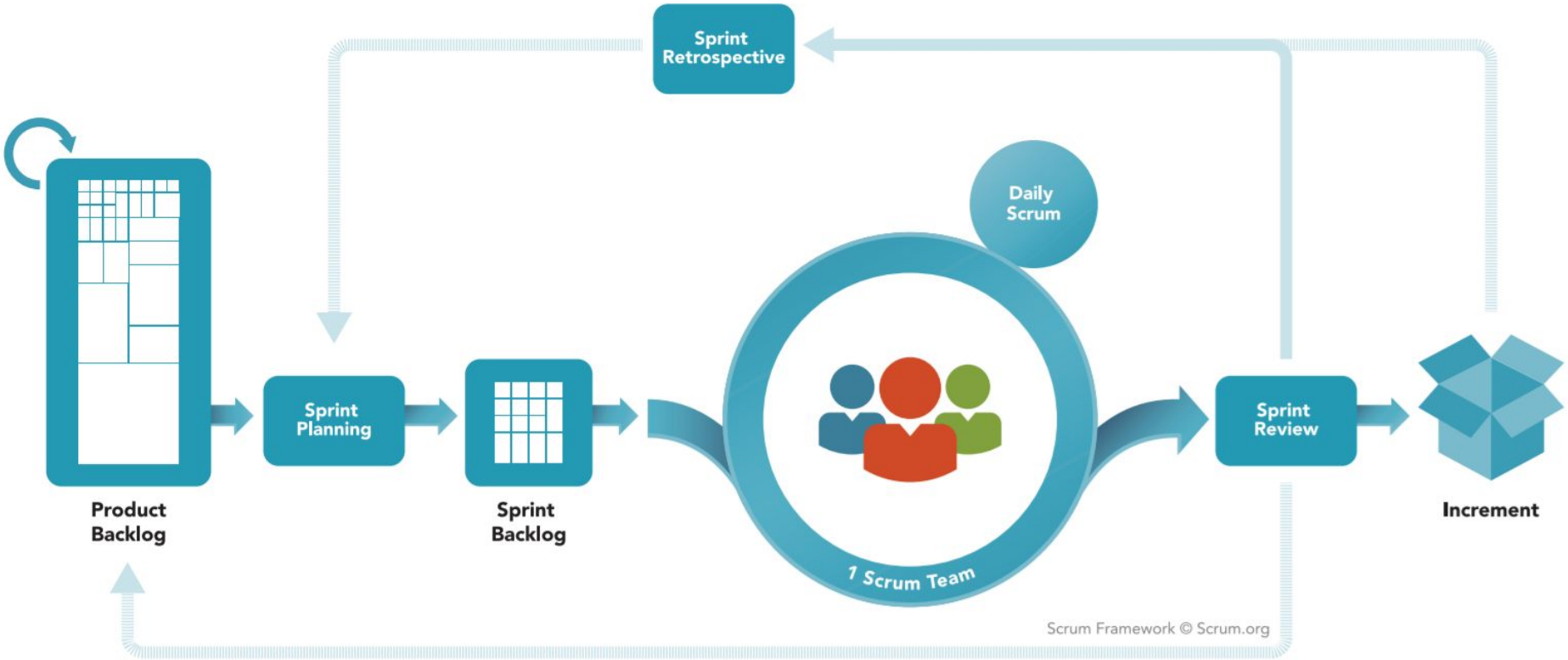
Highsmith's Agile Development Framework





What it is to scrum

https://www.scrum.org/resources/what-is-scrum/


Sprint Retrospective
Daily Scrum
Product Backlog
Sprint Planning
Sprint Backlog
1 Scrum Team
Sprint Review
Increment
Scrum Framework © Scrum.org

Scrum.org

But first....

Research paper discussion!

Don't forget to check on next week's!

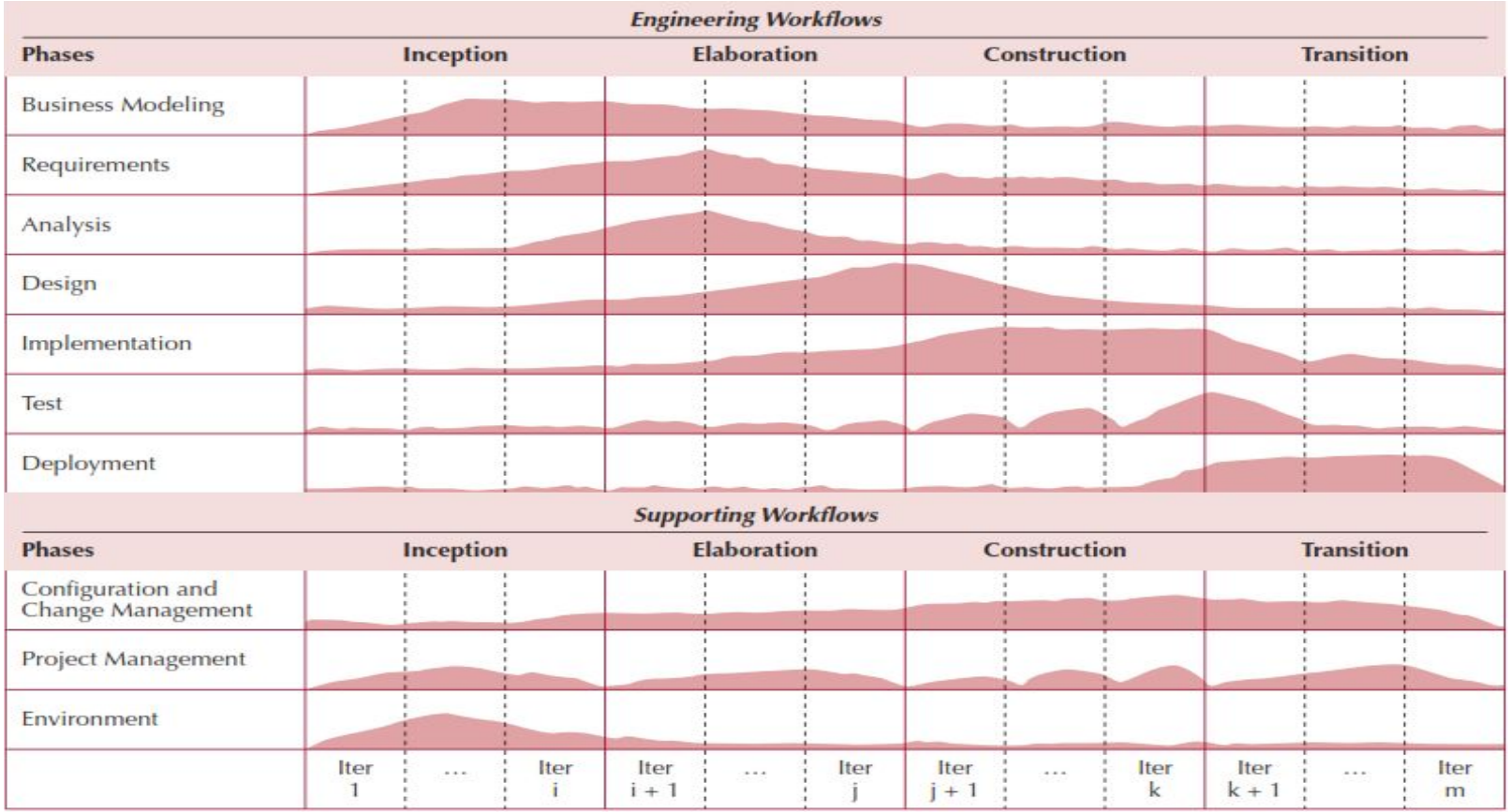
Which methodology?

Ability to Develop Systems	Structured Methodologies		RAD Methodologies			Agile Methodologies	
	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP	SCRUM
With Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent	Excellent
With Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Good	Good
That Are Complex	Good	Good	Good	Poor	Excellent	Good	Good
That Are Reliable	Good	Good	Good	Poor	Excellent	Excellent	Excellent
With a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent	Excellent
With Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Excellent	Excellent

Moreover, which methodology **and why?**

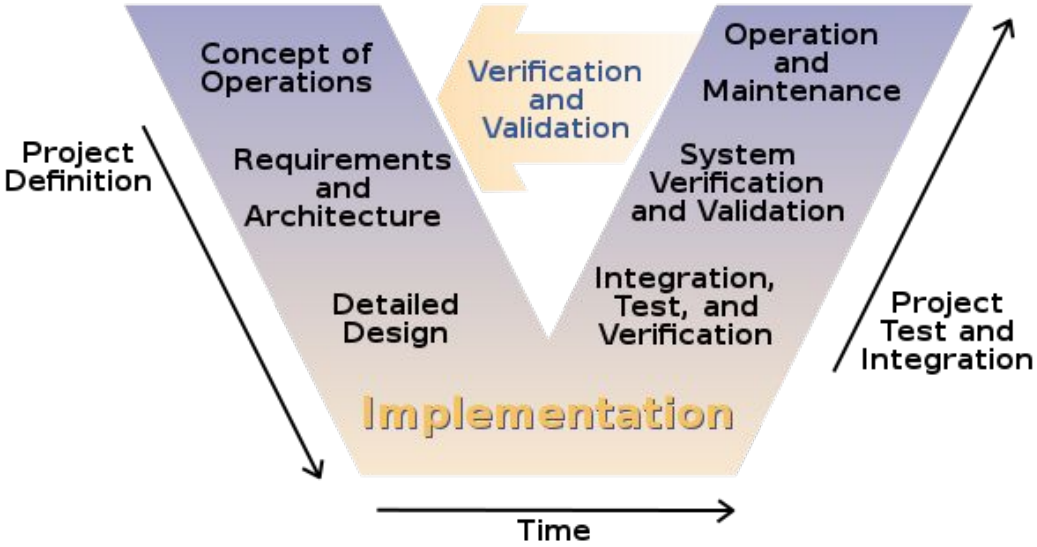
- You have to pick one at the beginning of your project!
- Consider this a spectrum (left to right)

The Unified Process

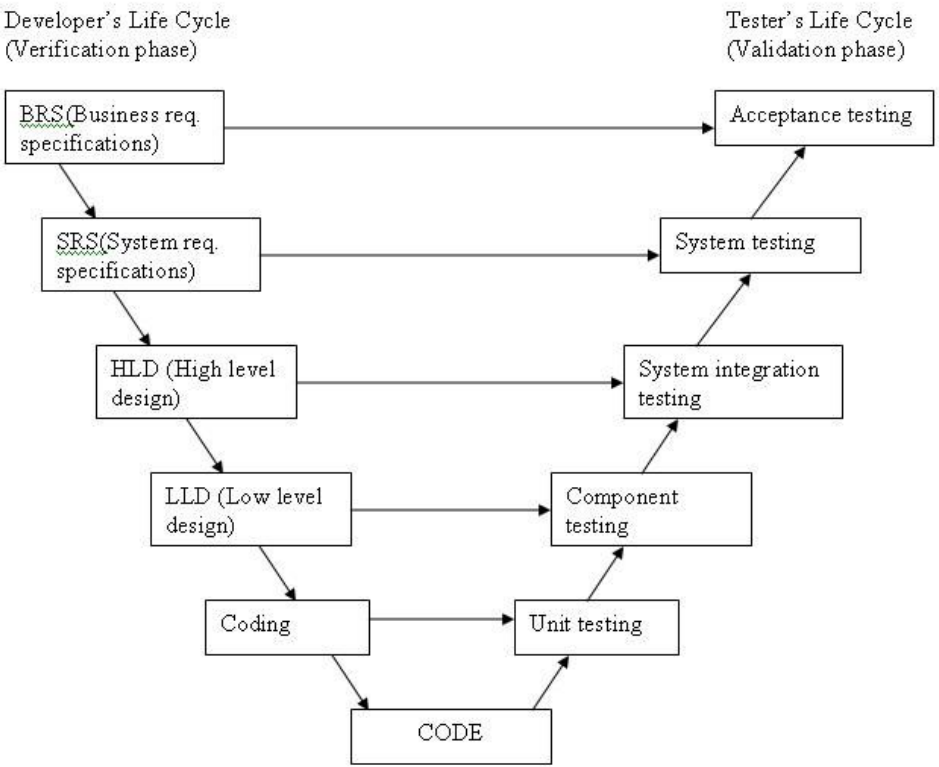


So we have learned a bit about a few different approaches

We've touched on waterfall, some rapid prototyping, agile (plus scrum)
There's also the V-model!



*Clarus Concept of Operations. Publication No. FHWA-JPO-05-072, Federal Highway Administration (FHWA), 2005



* http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/

Time for another discussion!

Marcus Weber, IS project manager at ICAN Mutual Insurance Co., is reviewing the staffing arrangements for his next major project, the development of an expert system-based underwriter's assistant. This new system will involve a whole new way for the underwriters to perform their tasks.

The underwriter's assistant system will function as sort of an underwriting supervisor, reviewing key elements of each application, checking for consistency in the underwriter's decisions, and ensuring that no critical factors have been overlooked. The goal of the new system is to improve the quality of the underwriters' decisions and to improve underwriters' productivity. It is expected that the new system will substantially change the way the underwriting staff do their jobs.

Marcus is dismayed to learn that because of budget constraints, he must choose between one of two available staff members. Barry Filmore has had considerable experience and training in individual and organizational behavior. Barry has worked on several other projects in which the end users had to make significant adjustments to the new system, and Barry seems to have a knack for anticipating problems and smoothing the transition to a new work environment. Marcus had hoped to have Barry's involvement in this project.

Marcus's other potential staff member is Kim Danville. Prior to joining ICAN Mutual, Kim had considerable work experience with the expert system technologies that ICAN has chosen for this expert system project. Marcus was counting on Kim to help integrate the new expert system technology into ICAN's systems environment, and also to provide on-the-job training and insights to the other developers on this team.

Given that Marcus's budget will only permit him to add Barry or Kim to this project team, but not both, what choice do you recommend for him? Justify your answer

Take ~5-10 minutes and discuss again

Who would you add to the team and **why**?

~~(Make sure you write down your answer **and** the names of your teammates)~~
~~(You'll submit this, plus effort later, to Blackboard after class)~~

PROJECT MANAGEMENT

(WordArt/Comic Sans intended)

About the homework...

1) It is due soon!
2) I'll give you some time to work on it as a group tonight

at ~8:30 (S1)
or ~5:00 (S2)

Project management

Process of planning and controlling system development within a specified time at a minimum cost with the right functionality

Project:

- Set of activities with a specified beginning and end point meant to create a system that brings value to the business

Project Managers:

- Monitor and control all tasks and roles that need to be coordinated

Project management

Inception phase:

- Generate a **system request** based on a business need or opportunity

Perform a feasibility analysis

- Revise the system request
- Approve or decline the project

Project identification

Projects are driven by business needs

- Identified by business people
- Identified by IT people
- (better yet) identified jointly by business and IT

Project sponsor believes in the system and wants to see it succeed

- Normally this is a business person
- Should have the authority to move it forward
- May also be called a "champion"

Business value

Tangible Value

- Can be quantified and measured directly
- Example: 2 percent reduction in operating costs

Intangible Value

- We know it will add value & save time, but we may not be able to quantify or measure its benefits
- Example: improved customer service

System request

Document that describes the reasons for and the value added from building a new system

Contains 5 elements:

- **Project sponsor**: the primary point of contact for the project
- **Business need**: the reason prompting the project
- **Business requirements**: what the system will do
- **Business value**: how will the organization benefit from the project
- **Special issues**: anything else that should be considered

Feasibility analysis

Is this project **feasible**?

- What are the risks?
- Can these risks be overcome?

Major components:

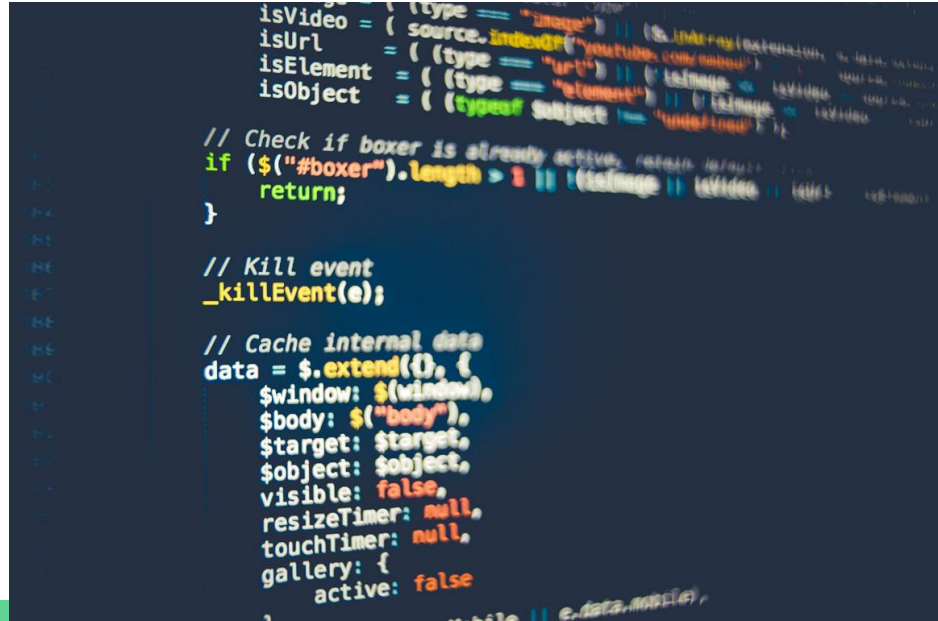
- Technical feasibility (Can we build it?)
- Economic feasibility (Should we build it?)
- Organizational feasibility (Will they use it?)

https://www.youtube.com/watch?v=WI6_snOjlm0

Technical feasibility

Identify risks in the following areas:

- The functional area:
  - Are analysts familiar with this portion of the business?
- The technology:
  - Less familiarity generates more risk
- Project size:
  - Large projects have more risk
- Compatibility:
  - Difficult integration increases the risk



Cost-benefit analysis

Identify the costs and the benefits

Assign values to the costs and benefits

Determine the cash flow

Determine the value using one or more methods:

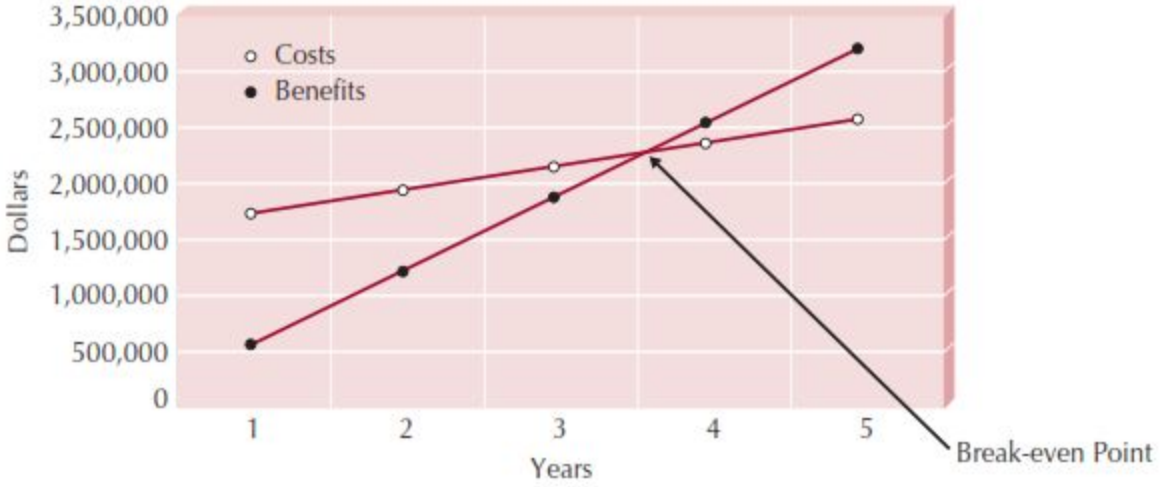
- Net present value (NPV)
- Return on investment (ROI)
- Break-even point

Formulas for determining value

Calculation	Definition	Formula
Present Value (PV)	The amount of an investment today compared to that same amount in the future, taking into account inflation and time.	$\frac{\text{Amount}}{(1 + \text{interest rate})^n}$ n = number of years in future
Net Present Value (NPV)	The present value of benefit less the present value of costs.	PV Benefits − PV Costs
Return on Investment (ROI)	The amount of revenues or cost savings results from a given investment.	$\frac{\text{Total benefits} - \text{Total costs}}{\text{Total costs}}$
Break-Even Point	The point in time at which the costs of the project equal the value it has delivered.	$\frac{\text{Yearly NPV*} - \text{Cumulative NPV}}{\text{Yearly NPV*}}$ *Use the Yearly NPV amount from the first year in which the project has a positive cash flow. Add the above amount to the year in which the project has a positive cash flow.

Formulas for determining value (break-even)

Calculation	Definition	Formula
Present Value (PV)	The amount of an investment today compared to that same amount in the future.	Amount
Net Present Value (NPV)		future sts
Return on Investment (RO		costs
Break-Even Point		e NPV



*Use the Yearly NPV amount from the first year in which the project has a positive cash flow.

Add the above amount to the year in which the project has a positive cash flow.

CBA Example

	2008	2009	2010	2011	2012	Total
Increased sales	500,000	530,000	561,800	595,508	631,238	
Reduction in customer complaint calls	70,000	70,000	70,000	70,000	70,000	
Reduced inventory costs	68,000	68,000	68,000	68,000	68,000	
TOTAL BENEFITS:	638,000	668,000	699,800	733,508	769,238	
PV OF BENEFITS:	619,417	629,654	640,416	651,712	663,552	3,204,752
PV OF ALL BENEFITS:	619,417	1,249,072	1,889,488	2,541,200	3,204,752	
2 Servers @ $125,000	250,000	0	0	0	0	
Printer	100,000	0	0	0	0	
Software licenses	34,825	0	0	0	0	
Server software	10,945	0	0	0	0	
Development labor	1,236,525	0	0	0	0	
TOTAL DEVELOPMENT COSTS:	1,632,295	0	0	0	0	
Hardware	54,000	81,261	81,261	81,261	81,261	
Software	20,000	20,000	20,000	20,000	20,000	
Operational labor	111,788	116,260	120,910	125,746	130,776	
TOTAL OPERATIONAL COSTS:	185,788	217,521	222,171	227,007	232,037	
TOTAL COSTS:	1,818,083	217,521	222,171	227,007	232,037	
PV OF COSTS:	1,765,129	205,034	203,318	201,693	200,157	2,575,331
PV OF ALL COSTS:	1,765,129	1,970,163	2,173,481	2,375,174	2,575,331	
TOTAL PROJECT BENEFITS – COSTS:	(1,180,083)	450,479	477,629	506,501	537,201	
YEARLY NPV:	(1,145,712)	424,620	437,098	450,019	463,395	629,421
CUMULATIVE NPV:	(1,145,712)	(721,091)	(283,993)	166,026	629,421	
RETURN ON INVESTMENT:	24.44%	(629,421/2,575,331)				
BREAK-EVEN POINT:	3.63 years	[break-even occurs in year 4; (450,019 – 166,026)/450,019 = 0.63]				
INTANGIBLE BENEFITS:	This service is currently provided by competitors Improved customer satisfaction					

Organizational Feasibility

Will the users accept the system?

Is the project strategically aligned with the business?

Conduct a stakeholder analysis

- Project champion(s)
- Organizational management
- System users
- Others

	Role	Techniques for Improvement
Champion	A champion: • Initiates the project • Promotes the project • Allocates his or her time to project • Provides resources	• Make a presentation about the objectives of the project and the proposed benefits to those executives who will benefit directly from the system • Create a prototype of the system to demonstrate its potential value
Organizational Management	Organizational managers: • Know about the project • Budget enough money for the project • Encourage users to accept and use the system	• Make a presentation to management about the objectives of the project and the proposed benefits • Market the benefits of the system using memos and organizational newsletters • Encourage the champion to talk about the project with his or her peers
System Users	Users: • Make decisions that influence the project • Perform hands-on activities for the project • Ultimately determine whether the project is successful by using or not using the system	• Assign users official roles on the project team • Assign users specific tasks to perform with clear deadlines • Ask for regular feedback from users (e.g., at weekly meetings)

Initiation - developing a business case

System Request—Integrated Health Clinic Delivery System

Project sponsor: Max Ross, Vice President of Pharmacy Services

Business Need: This project has been initiated to integrate health clinic services by providing real-time electronic communication and scheduling for Patterson Superstore health clinics.

Business Requirements:

- Mobile appointment scheduling
- Real-time communication with medical personnel (audio, video, and text)
- Tele-health assessment and diagnosis of minor problems through video house calls
- Data analytic and tracking capabilities

Business Value:

We expect this integrated health clinic delivery system to lead to improved customer satisfaction and increased brand recognition due to its first mover advantage and increased convenience for clinic clients. Implementation of this system is also expected to boost in-pharmacy sales due to increased foot traffic in stores.

Conservative estimates of tangible value to the company per clinic include:

- $375,000 (75 percent of $500,000) in clinic services from new customers
- $750,000 (75 percent of $1,000,000) in clinic services from existing customers
- $50,000 in pharmacy sales from increase foot traffic in stores

Special Issues or Constraints:

- The Pharmacy Department views this as a strategic system that will add value to the current business model and will also provide customers with increased convenience and satisfaction.
- In order to gain first mover advantage, the system should be implemented in phases with the appointment scheduling piece in place within six months from the approval date.
- Increased staffing will be needed to operate the new system from both the technical and business operations aspect.

Feasibility Analysis

Technical Feasibility: Can We Build It?

- Familiarity with Functional area: Less familiarity generates more risk
- Familiarity with Technology: Less familiarity generates more risk
- Project Size: Large projects have more risk
- Compatibility: The harder it is to integrate the system with the company's existing technology, the higher the risk

Economic Feasibility: Should We Build It?

- Development costs
- Annual operating costs
- Annual benefits (cost savings and revenues)
- Intangible costs and benefits

Organizational Feasibility: If We Build It, Will They Come?

- Is the project strategically aligned with the business?
- Project champion(s)
- Senior management
- Users
- Other stakeholders

Economic Feasibility

1. Identifing Costs and Benefits	List the tangible costs and benefits for the project. Include both one-time and recurring costs.
2. Assigning Values to Costs and Benefits	Work with business users and IT professionals to create numbers for each of the costs and benefits. Even intangibles should be valued if at all possible.
3. Determining Cash Flow	Project what the costs and benefits will be over a period of time, usually three to five years. Apply a growth rate to the numbers, if necessary.
4. Determining Net Present Value (NPV)	Calculate what the value of future costs and benefits are if measured by today's standards. You will need to select a rate of growth to apply the NPV formula.
5. Determining Return on Investment (ROI)	Calculate how much money the organization will receive in return for the investment it will make using the ROI formula.
6. Determining the Break-Even Point	Find the first year in which the system has greater benefits than costs. Apply the break-even formula using figures from that year. This will help you understand how long it will take before the system creates real value for the organization.
7. Graphing the Break-Even Point	Plot the yearly costs and benefits on a line graph. The point at which the lines cross is the break-even point.

Costs and Benefits

Development Costs	Operational Costs
Development Team Salaries	Software Upgrades
Consultant Fees	Software Licensing Fees
Development Training	Hardware Repairs
Hardware and Software	Hardware Upgrades
Vendor Installation	Operational Team Salaries
Office Space and Equipment	Communications Charges
Data Conversion Costs	User Training
Tangible Benefits	**Intangible Benefits**
Increased Sales	Increased Market Share
Reductions in Staff	Increased Brand Recognition
Reductions in Inventory	Higher Quality Products
Reductions in IT Costs	Improved Customer Service
Better Supplier Prices	Better Supplier Relations

Project Selection

Projects are approved, declined or delayed based on value added vs. risks

Size	What is the size? How many people are needed to work on the project?
Cost	How much will the project cost the organization?
Purpose	What is the purpose of the project? Is it meant to improve the technical infrastructure? Support a current business strategy? Improve operations? Demonstrate a new innovation?
Length	How long will the project take before completion? How much time will go by before value is delivered to the business?
Risk	How likely is it that the project will succeed or fail?
Scope	How much of the organization is affected by the system? A department? A division? The entire corporation?
Return on investment	How much money does the organization expect to receive in return for the amount the project costs?

Project portfolio management

- Goals:
  - Maximize cost/benefit ratio
  - Maintain an optimal mix of projects based on:
    - Risk
    - Size, cost & length of time to complete
    - Purpose, scope & business value
  - Limited resources require trade-offs
- Selected projects enter the project management process
  - Project portfolio management

Project management tools

Aids in creating workplans

Identify all tasks, their sequence and estimate the time to complete each one

Work breakdown structures (WBS): a hierarchy of tasks to identify:

- Duration of each task
- Current status of each task
- Task dependencies (shows which tasks must be completed before others can begin)

Gantt charts: horizontal bar chart that shows the WBS graphically

Network diagrams: (Program Evaluation & Review Technique) PERT and Critical Path Method (CPM)

[Project Name] Project Schedule

Gantt Chart Template © 2006-2018 by Vertex42.com.

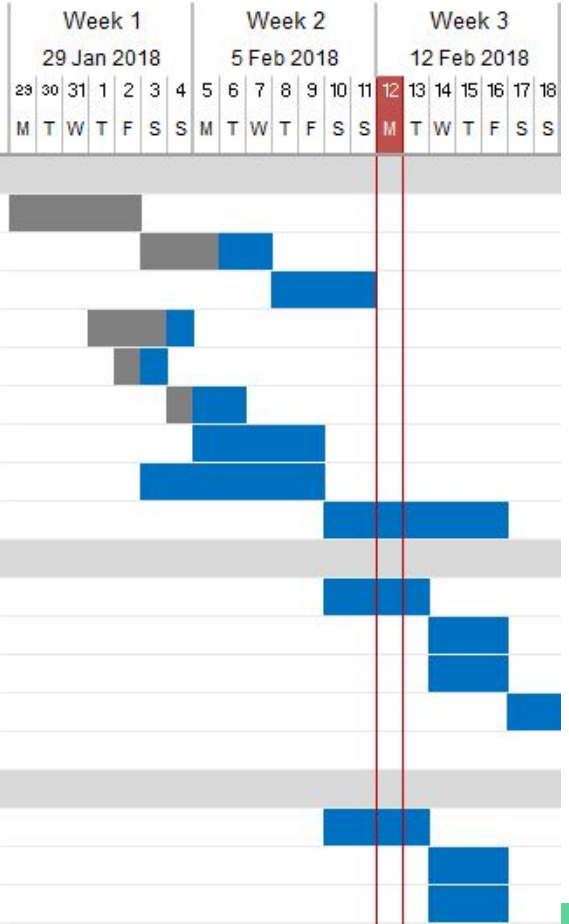
[Company Name]

Project Start Date: 1/29/2018 (Monday)

Display Week: 1

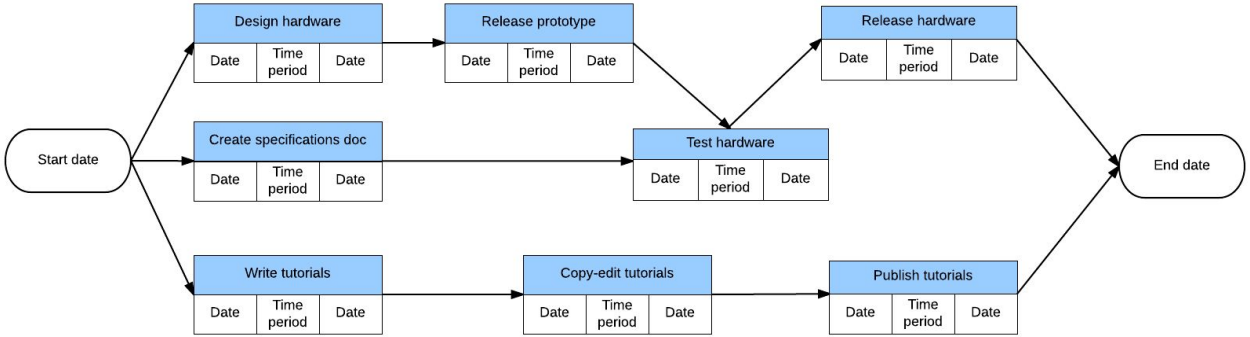
Project Lead: Ted Harris

WBS	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
1	[Task Category]			-			-
1.1	[Task]	[Name]	Mon 1/29/18	Fri 2/02/18	5	100%	5
1.2	[Task]		Sat 2/03/18	Wed 2/07/18	5	60%	3
1.3	[Task]		Thu 2/08/18	Sun 2/11/18	4	0%	2
1.4	[Task]		Thu 2/01/18	Sun 2/04/18	4	75%	2
1.4.1	[Sub-task]		Fri 2/02/18	Sat 2/03/18	2	50%	1
1.4.2	[Sub-task]		Sun 2/04/18	Tue 2/06/18	3	50%	2
1.5	[Task]		Mon 2/05/18	Fri 2/09/18	5	0%	5
1.6	[Task]		Sat 2/03/18	Fri 2/09/18	7	0%	5
1.7	[Task]		Sat 2/10/18	Fri 2/16/18	7	0%	5
2	[Task Category]			-			-
2.1	[Task]		Sat 2/10/18	Tue 2/13/18	4	0%	2
2.2	[Task]		Wed 2/14/18	Fri 2/16/18	3	0%	3
2.3	[Task]		Wed 2/14/18	Fri 2/16/18	3	0%	3
2.4	[Task]		Sat 2/17/18	Thu 2/22/18	6	0%	4
2.5	[Task]		Fri 2/23/18	Sun 2/25/18	3	0%	1
3	[Task Category]			-			-
3.1	[Task]		Sat 2/10/18	Tue 2/13/18	4	0%	2
3.2	[Task]		Wed 2/14/18	Fri 2/16/18	3	0%	3
3.3	[Task]		Wed 2/14/18	Fri 2/16/18	3	0%	3



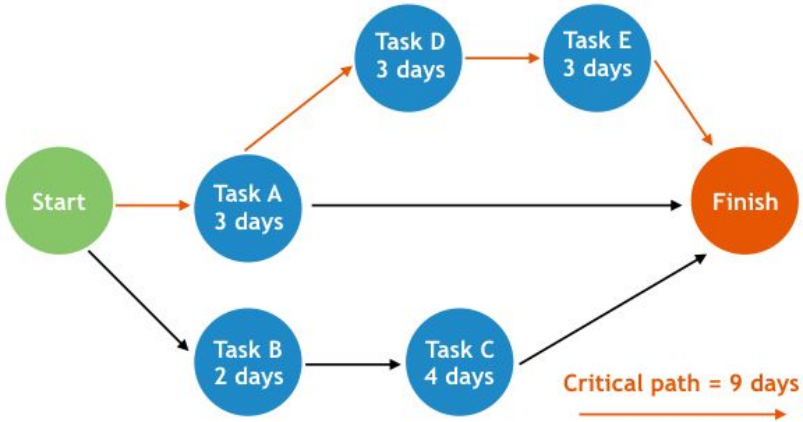
https://keydifferences.com/difference-between-pert-and-cpm.html

PERT and CPM



PERT

CPM



All of these are verboten.

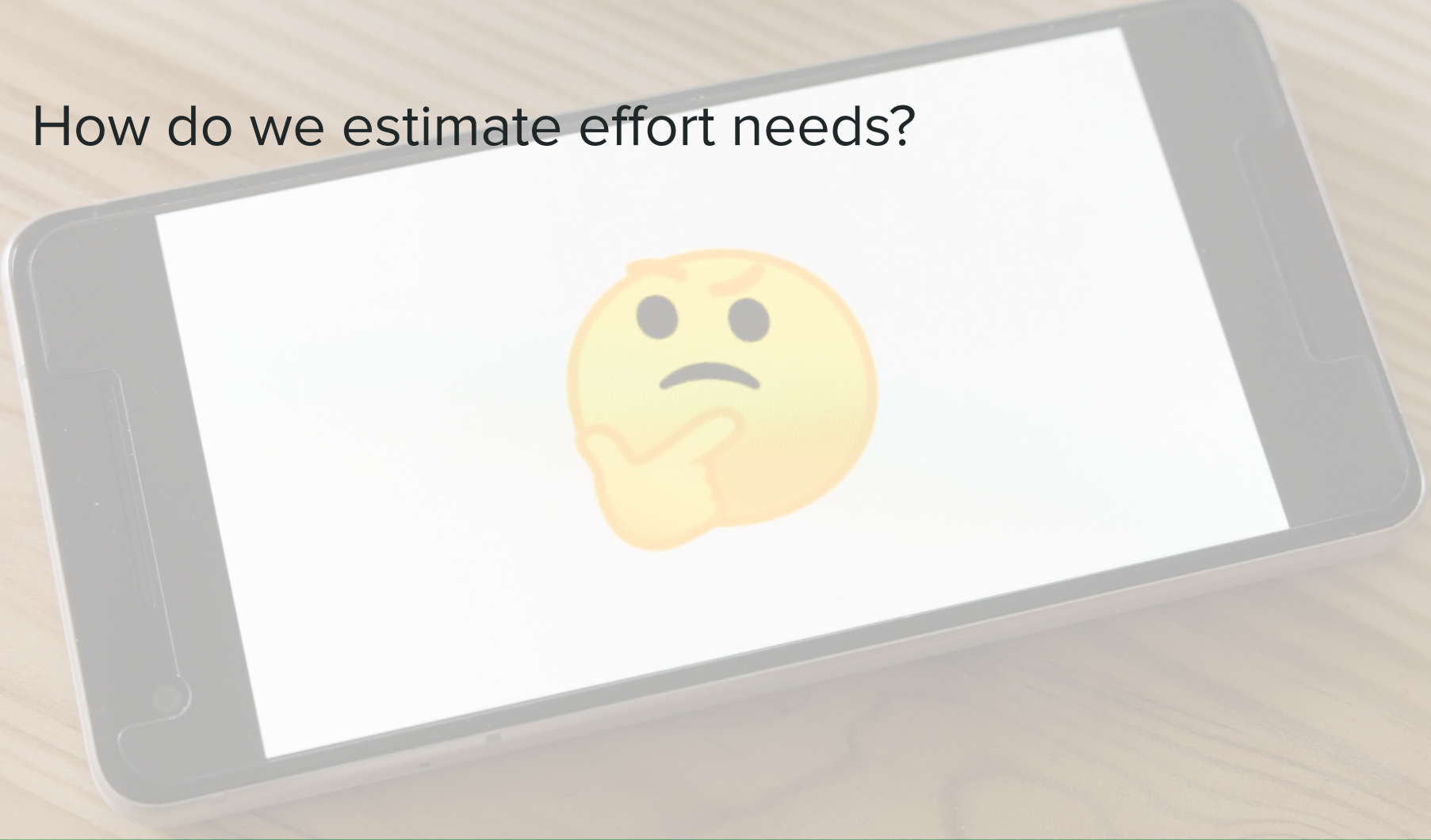
1. Directly copy and pasted from online source without attribution and without writing in your own words.

2. Downloading a solved problem from an internet source.
   a. Wording was rewritten, but the answer is nearly identical.

3. Individual assignments not performed individually.
   a. Your answers are nearly identical with minor word changes.

If it happens again, I will refer the matter to OSCCR (both the new instance AND the prior instance, regardless of what grade you received on the assignment).

How do we estimate effort needs?



Project effort estimation

Estimation involves trade-offs between functionality, time and cost

It is the process of assigning projected values for time and effort

Most accurate estimates come from experience

Use-case point method; based on:

- Technical complexity factors
- Environmental factors

Use case estimation example

Actors and use cases

Unadjusted Actor Weighting Table:

Actor Type	Description	Weighting Factor	Number	Result
Simple	External system with well-defined API	1	0	0
Average	External system using a protocol-based interface, e.g., HTTP, TCT/IP, or a database	2	0	0
Complex	Human	3	4	12
			Unadjusted Actor Weight Total (UAW)	**12**

Unadjusted Use-Case Weighting Table:

Use Case Type	Description	Weighting Factor	Number	Result
Simple	1–3 transactions	5	3	15
Average	4–7 transactions	10	4	40
Complex	>7 transactions	15	1	15
			Unadjusted Use Case Weight Total (UUCW)	**70**

Unadjusted Use-Case Points (UUCP) = UAW + UUCW **82 = 12 + 70**

Use case estimation example

Technical complexity

Technical Complexity Factors:

Factor Number	Description	Weight	Assigned Value (0 – 5)	Weighted Value	Notes
T1	Distributed system	2.0	0	0	
T2	Response time or throughput performance objectives	1.0	5	5	
T3	End-user online efficiency	1.0	3	3	
T4	Complex internal processing	1.0	1	1	
T5	Reusability of code	1.0	1	1	
T6	Ease of installation	0.5	2	1	
T7	Ease of use	0.5	4	2	
T8	Portability	2.0	0	0	
T9	Ease of change	1.0	2	2	
T10	Concurrency	1.0	0	0	
T11	Special security objectives included	1.0	0	0	
T12	Direct access for third parties	1.0	0	0	
T13	Special user training required	1.0	0	0	
			Technical Factor Value (TFactor)	**15**	

Technical Complexity Factor (TCF) = 0.6 + (0.01 * TFactor) **0.75 = 0.6 + (0.01 * 15)**

Use case estimation example

Environmental factors + final estimate

Environmental Factors:

Factor Number	Description	Weight	Assigned Value (0 – 5)	Weighted Value	Notes
E1	Familiarity with system development process being used	1.5	4	6	
E2	Application experience	0.5	4	2	
E3	Object-oriented experience	1.0	4	4	
E4	Lead analyst capability	0.5	5	2.5	
E5	Motivation	1.0	5	5	
E6	Requirements stability	2.0	5	10	
E7	Part-time staff	–1.0	0	0	
E8	Difficulty of programming language	–1.0	4	–4.0	
			Environmental Factor Value (EFactor)	**25.5**	

Environmental Factor (EF) = 1.4 + (−0.03 * EFactor) **0.635 = 1.4 + (−0.03 * 25.5)**

Adjusted Use Case Points (UCP) = UUCP * TCF * ECF **33.3375 = 70 * 0.75 * 0.635**

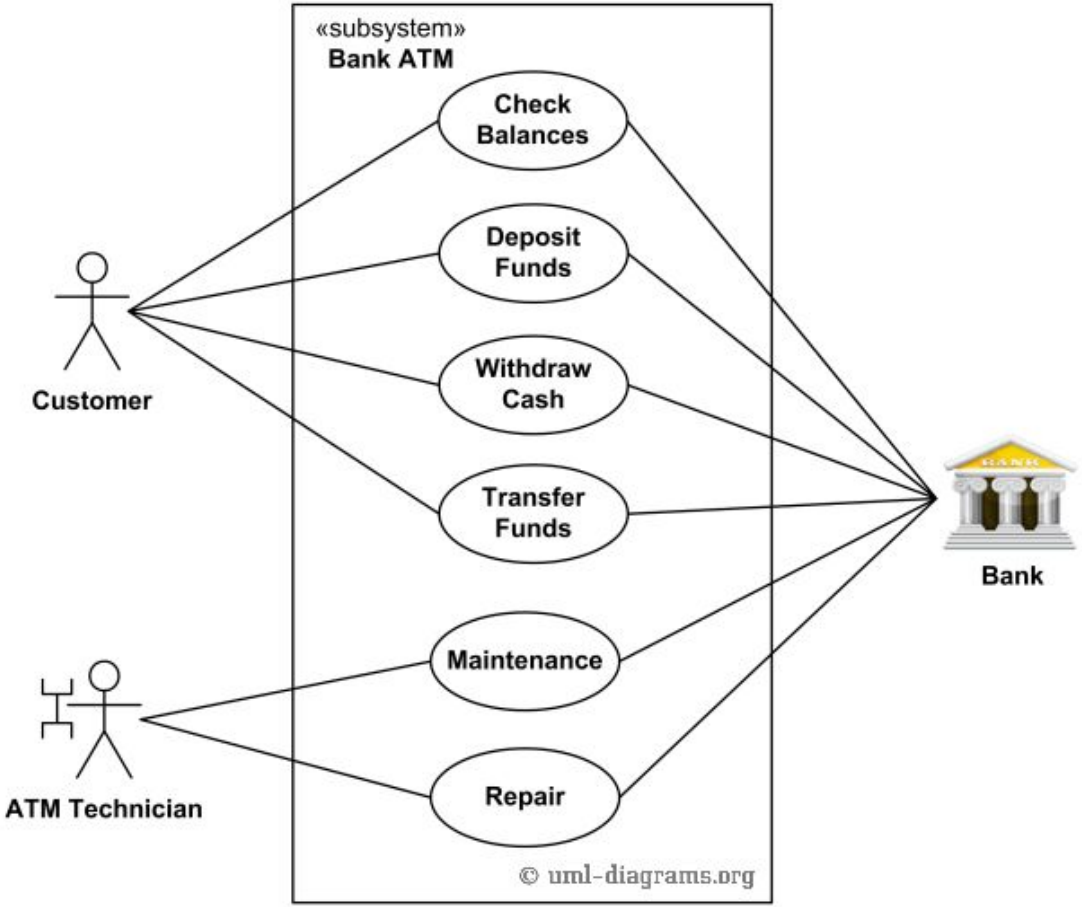
Effort in person-hours = UCP * PHM **666.75 = 20 * 33.3375**

Use Case Points Analysis

How many actors?

How many use cases?

How much effort per use case?



Planning Tasks & WBS

Workplan Information	Example
Name of the task	Perform economic feasibility
Start date	Jan 05, 2015
Completion date	Jan 19, 2015
Person assigned to the task	Project sponsor: Mary Smith
Deliverable(s)	Cost-benefit analysis
Completion status	Open
Priority	High
Resources that are needed	Spreadsheet software
Estimated time	16 hours
Actual time	14.5 hours

I. Business Modeling
- **a.** Inception
- **b.** Elaboration
- **c.** Construction
- **d.** Transition
- **e.** Production

II. Requirements
- **a.** Inception
- **b.** Elaboration
- **c.** Construction
- **d.** Transition
- **e.** Production

III. Analysis
- **a.** Inception
- **b.** Elaboration
- **c.** Construction
- **d.** Transition
- **e.** Production

IV. Design
- **a.** Inception
- **b.** Elaboration
- **c.** Construction

V. Implementation
- **a.** Inception
- **b.** Elaboration
- **c.** Construction
- **d.** Transition
- **e.** Production

VI. Test
- **a.** Inception
- **b.** Elaboration
- **c.** Construction
- **d.** Transition
- **e.** Production

VII. Deployment
- **a.** Inception
- **b.** Elaboration
- **c.** Construction
- **d.** Transition
- **e.** Production

VIII. Configuration and Change Management
- **a.** Inception
- **b.** Elaboration
- **c.** Construction
- **d.** Transition
- **e.** Production

IX. Project Management
- **a.** Inception
- **b.** Elaboration
- **c.** Construction
- **d.** Transition
- **e.** Production

X. Environment
- **a.** Inception
- **b.** Elaboration
- **c.** Construction
- **d.** Transition
- **e.** Production

XI. Operations and Support
- **a.** Inception
- **b.** Elaboration
- **c.** Construction
- **d.** Transition
- **e.** Production

XII. Infrastructure Management
- **a.** Inception
- **b.** Elaboration
- **c.** Construction
- **d.** Transition
- **e.** Production

Task Number	Task Name	Duration (in weeks)	Dependency	Status
1	Identify vendors	2		Complete
2	Review training materials	6	1	Complete
3	Compare vendors	2	2	In Progress
4	Negotiate with vendors	3	3	Open
5	Develop communications information	4	1	In Progress
6	Disseminate information	2	5	Open
7	Create and administer survey	4	6	Open
7.1	Create initial survey	1		Open
7.2	Review initial survey	1	7.1	Open
7.2.1	Review by Director of IT Training	1		Open
7.2.2	Review by Project Sponsor	1		Open
7.2.3	Review by Representative Trainee	1		Open
7.3	Pilot test initial survey	1	7.1	Open
7.4	Incorporate survey changes	1	7.2, 7.3	Open
7.5	Create distribution list	0.5		Open
7.6	Send survey to distribution list	0.5	7.4, 7.5	Open
7.7	Send follow-up message	0.5	7.6	Open
7.8	Collect completed surveys	1	7.6	Open
8	Analyze results and choose vendor	2	4, 7	Open
9	Build new classrooms	11	1	In Progress
10	Develop course options	3	8, 9	Open

Creating/managing the **workplan**

Workplan:
- Dynamic and sequential list of all tasks needed to complete a project

Approaches:
- Modify existing or completed projects
- Derive the tasks from the methodology being used

Unified Process:
- Iterative and incremental
- Workplan is also iterative and incremental
- Tasks and time intervals follow the phases
- Different tasks executed for each workflow

Scope management

Scope **creep**

- Occurs after the project is underway
- Results from adding new requirements to the project
- Can have a deleterious effect on the schedule

Techniques to manage the project scope:

- Identify all requirements at the outset
- Allow only those changes deemed absolutely necessary
- Carefully examine the impact of suggested changes
- Delay some changes for “future enhancements”
- Time boxing

Execution and Control

Develop workplans – **but plans change**

Change Management

Timeboxing – “not about time”

Refine Estimates:

Assumptions	Actions	Level of Risk
If you assume the rest of the project is simpler than the part that was late and is also simpler than believed when the original schedule estimates were made, you can make up lost time.	Do not change schedule.	High risk
If you assume the rest of the project is simpler than the part that was late and is no more complex than the original estimate assumed, you can't make up the lost time, but you will not lose time on the rest of the project.	Increase the entire schedule by the total amount of time that you are behind (e.g., if you missed the scheduled date by two weeks, move the rest of the schedule dates to two weeks later). If you included padded time at the end of the project in the original schedule, you might not have to change the promised system delivery date; you'll just use up the padded time.	Moderate risk
If you assume that the rest of the project is as complex as the part that was late (your original estimates were too optimistic), then all the scheduled dates in the future underestimate the real time required by the same percentage as the part that was late.	Increase the entire schedule by the percentage of weeks that you are behind (e.g., if you are two weeks late on part of the project that was supposed to take eight weeks, you need to increase all remaining time estimates by 25 percent). If this moves the new delivery date beyond what is acceptable to the project sponsor, the scope of the project must be reduced.	Low risk

How would you manage the following?

1) Your project manager throws a new team member at you to "help with the effort." How do you manage this change?

2) You are in danger of going over budget and must cut efforts. Do you minimize time spent on verification/validation or on features requested by the customer?

3) You go to an industry conference (probably with 4.0 or 5.0 in the title somewhere) and learn about a new business process that is a "game changer." Your manager (me) is hesitant because he is a curmudgeon however you feel that it would be beneficial for your project.

RISK
Opdrachten
RISK
Territories
GOED GEZAPT

Risk Management

Risk Assessment

RISK 1:	The development of this system likely will be slowed considerably because project team members have not programmed in Java prior to this project.
Likelihood of risk:	High probability of risk.
Potential impact on the project:	This risk will probably increase the time to complete programming tasks by 50 percent.

Ways to address this risk:

It is very important that time and resources are allocated to up-front training in Java for the programmers who are used for this project. Adequate training will reduce the initial learning curve for Java when programming begins. Additionally, outside Java expertise should be brought in for at least some part of the early programming tasks. This person should be used to provide experiential knowledge to the project team so that Java-related issues (of which novice Java programmers would be unaware) are overcome.

RISK 2: ...

What are some common risks for a project?

Staffing

Goals:

- Determine how many people are required
- Match skill sets to required activities
- Motivate the team to meet the objectives
- Minimize conflicts

Deliverable ➔ **The staffing plan**

- Number and kind of people assigned
- Overall reporting structure
- Project charter (describes the project's objectives and rules)

Creating a cohesive team (jellin') ... AKA a PM's dream

A team of people so strongly knit that the whole is greater than the sum of its parts

Characteristics of a jelled team:

- Very low turnover rate
- Strong sense of identity
- A feeling of eliteness
- Team vs. individual ownership of the project
- Team members enjoy their work

Staffing plan

Calculate the number of people needed:

$$number\ of\ people = \frac{person\text{-}months}{time\ to\ complete\ (in\ months)}$$

- Lines of communication increase exponentially as people are added
- Create a reporting structure for projects with large numbers of people assigned
- Form sub-teams as necessary
- Assign the Project Manager, Functional lead & Technical lead
- Pay attention to technical and interpersonal skills

Motivating your team

Motivation is the greatest influence on performance
Monetary rewards usually do not motivate

Suggested motivating techniques:

- 20% time rule
- Peer-to-peer recognition awards
- Team ownership (refer to the team as “we”)
- Allow members to focus on what interests them
- Utilize equitable compensation
- Encourage group ownership
- Provide for autonomy, but trust the team to deliver

Conflict!

Preventing or mitigating conflict:

- Cohesiveness has the greatest effect
- **Clearly defining roles** and holding team members accountable
- Establish work & communications rules in the project charter

Additional techniques:

- **Clearly define plans** for the project
- Make sure the team **understands** the importance of the project
- Develop detailed operating procedures
- Develop a project charter
- Develop a schedule of commitments in advance
- Forecast other priorities and their impact on the project

On the docket...

1. New homework / old homework
2. GitHub Project
3. Research paper presentation

Environment/Infrastructure management

Environment ➔ choose the right set of tools

Use appropriate CASE tools to:

- Increase productivity and centralize information (repository)
- Utilize diagrams—more easily understood

Establish standards to reduce complexity

Infrastructure—Document the project appropriately

- Store deliverables/communications in a project binder (or appropriate repository)
- Use Unified Process standard documents
- Don’t put off documentation to the last minute

Group Exercise – ~15 minutes

Pretend that your term project is a real project.

Prepare a sample system request for it.

(Spend 5 minutes spitballing ideas and 10 minutes coming up with a req)

System Request—Integrated Health Clinic Delivery System

Project sponsor: Max Ross, Vice President of Pharmacy Services

Business Need: This project has been initiated to integrate health clinic services by providing real-time electronic communication and scheduling for Patterson Superstore health clinics.

Business Requirements:

- Mobile appointment scheduling
- Real-time communication with medical personnel (audio, video, and text)
- Tele-health assessment and diagnosis of minor problems through video house calls
- Data analytic and tracking capabilities

Business Value:

We expect this integrated health clinic delivery system to lead to improved customer satisfaction and increased brand recognition due to its first mover advantage and increased convenience for clinic clients. Implementation of this system is also expected to boost in-pharmacy sales due to increased foot traffic in stores.

Conservative estimates of tangible value to the company per clinic include:

- $375,000 (75 percent of $500,000) in clinic services from new customers
- $750,000 (75 percent of $1,000,000) in clinic services from existing customers
- $50,000 in pharmacy sales from increase foot traffic in stores

Special Issues or Constraints:

- The Pharmacy Department views this as a strategic system that will add value to the current business model and will also provide customers with increased convenience and satisfaction.
- In order to gain first mover advantage, the system should be implemented in phases with the appointment scheduling piece in place within six months from the approval date.
- Increased staffing will be needed to operate the new system from both the technical and business operations aspect.

System request

Document that describes the reasons for and the value added from building a new system

Contains 5 elements:

- **Project sponsor**: the primary point of contact for the project
- **Business need**: the reason prompting the project
- **Business requirements**: what the system will do
- **Business value**: how will the organization benefit from the project
- **Special issues**: anything else that should be considered

BEGIN.

(unsplash's only picture of 'requirement')

https://www.youtube.com/watch?v=BKorP55Aqvg

Next week ➜ sign up!

Now let's start into analysis: **Requirements**

WHAT IS A **REQUIREMENT**?

A statement of what the system must do or a characteristic it must have.

Requirements intro

The systems development process transforms the existing (as is) system into the proposed (to be) system

Requirements determination

- The single most critical step of the entire SDLC
- Changes can be made easily in this stage
- Most (>50%) system failures are due to problems with requirements
- The iterative process of OOSAD is effective because:
  - Small batches of requirements can be identified and implemented incrementally
  - The system will evolve over time

Good Requirements
(IEEE830 has a slightly different list)

VERIFIABLE – can be tested/possible to evaluate

CLEAR AND CONCISE – singular, short (<50 words), unambiguous, closed-ended and objective

COMPLETE – contains all info needed, leaves no one guessing, includes measurement units

CONSISTENT – no conflicts, same terminology throughout, does not duplicate

TRACEABLE – unique identity, cannot be separated, easily be traced, change control

VIABLE – realistic, met using existing tech, achieved w/in budget and schedule, have the skills to utilize, will be used, helpful to build the system

NECESSARY – necessary to meet objectives, absolutely critical, deficiency if removed

IMPLEMENTATION FREE – defines what functions are provided, does not specify how, allows developers to select tech best suited.

Are these **good** requirements?

- The system shall be user-friendly
- All screens must appear on the monitor quickly
- On loss of power, the battery backup must support normal operations.
- The electronic batch records must be Part 11 compliant
- Training for 21 CFR part 11 must be established.
- The system must generate a batch end report and a discrepancy report when a batch is aborted
- ...shall be installed with no disruption to production
- All PCs must have the same configuration
- After 3 unsuccessful login attempts, a javascript routine must run to lock the user out of the system.

Functional vs. non-functional requirements

FUNCTIONAL – process to perform or information it must contain

- FUNCTIONAL QUALITY – how much the actual problem is solved by the system
- NON-FUNCTIONAL QUALITY - efficiency, maintainability, portability, reliability, reusability, testability, usability

NON-FUNCTIONAL – behavioral properties the system must have

- Operational
- Performance
- Security
- Cultural and political

Requirements

Functional & non-functional requirements listed in outline format

May be prioritized

Provides information needed in subsequent workflows

Defines the scope of the system

Nonfunctional Requirements

1. Operational Requirements
1.1. The system will operate in Windows environment.
1.2. The system should be able to connect to printers wirelessly.
1.3. The system should automatically back up at the end of each day.

2. Performance Requirements
2.1. The system will store a new appointment in 2 seconds or less.
2.2. The system will retrieve the daily appointment schedule in 2 seconds or less.

3. Security Requirements
3.1. Only doctors can set their availability.
3.2. Only a manager can produce a schedule.

4. Cultural and Political Requirements
4.1. No special cultural and political requirements are anticipated.

Functional Requirements

1. Manage Appointments
1.1. Patient makes new appointment.
1.2. Patient changes appointment.
1.3. Patient cancels appointment.

2. Produce Schedule
2.1. Office Manager checks daily schedule.
2.2. Office Manager prints daily schedule.

3. Record Doctor Availability
3.1. Doctor updates schedule

Requirements definition

Determine the types of functional and non-functional requirements applicable to the project

Use requirements-gathering techniques to collect details

Analysts work with users to verify, change and prioritize each requirement

Continue this process through analysis workflow, but be careful of scope creep

Requirements that meet a need but are not within the current scope can be added to a list of future enhancements

A short interlude!

Now, if you search for requirements on YT you will come across many definitions, strategies, etc.

- Some of which may be *cynical, fyi*

Here's a nice short one that encapsulates the highlights:
https://www.youtube.com/watch?v=j2nC1Tr1tyM

Problems!

Analyst may not have access to the correct users

Requirements specifications may be inadequate

Some requirements may not be known in the beginning

Verifying and validating requirements can be difficult

Analysis strategies

Problem analysis

- Ask users to identify problems with the current system
- Ask users how they would solve these problems
- Good for improving efficiency or ease-of-use

Root cause analysis

- Focus is on the cause of a problem, not its solution
- Create a prioritized list of problems
- Try to determine their causes
- Once the causes are known, solutions can be developed

Analysis strategies

Duration analysis

- Determine the time required to complete each step in a business process
- Compare this to the total time required for the entire process
- Large differences suggest problems that might be solved by:
  - Integrating some steps together
  - Performing some steps simultaneously (in parallel)

Activity-based costing

- Same as duration analysis but applied to costs

Informal benchmarking

- Analyzes similar processes in other successful organizations

Analysis strategies

Outcome analysis

- What does the customer want in the end?

Technology analysis

- Apply new technologies to business processes & identify benefits

Activity elimination

- Eliminate each activity in a business process in a “force-fit” exercise

Requirements gathering techniques

Process is used to:

- Uncover all requirements (those uncovered late in the process are more difficult to incorporate)
- Build support and trust among users

Which technique(s) to use?

- Interviews
- Joint Application Development (JAD)
- Questionnaires
- Document analysis
- Observation

Interviewing

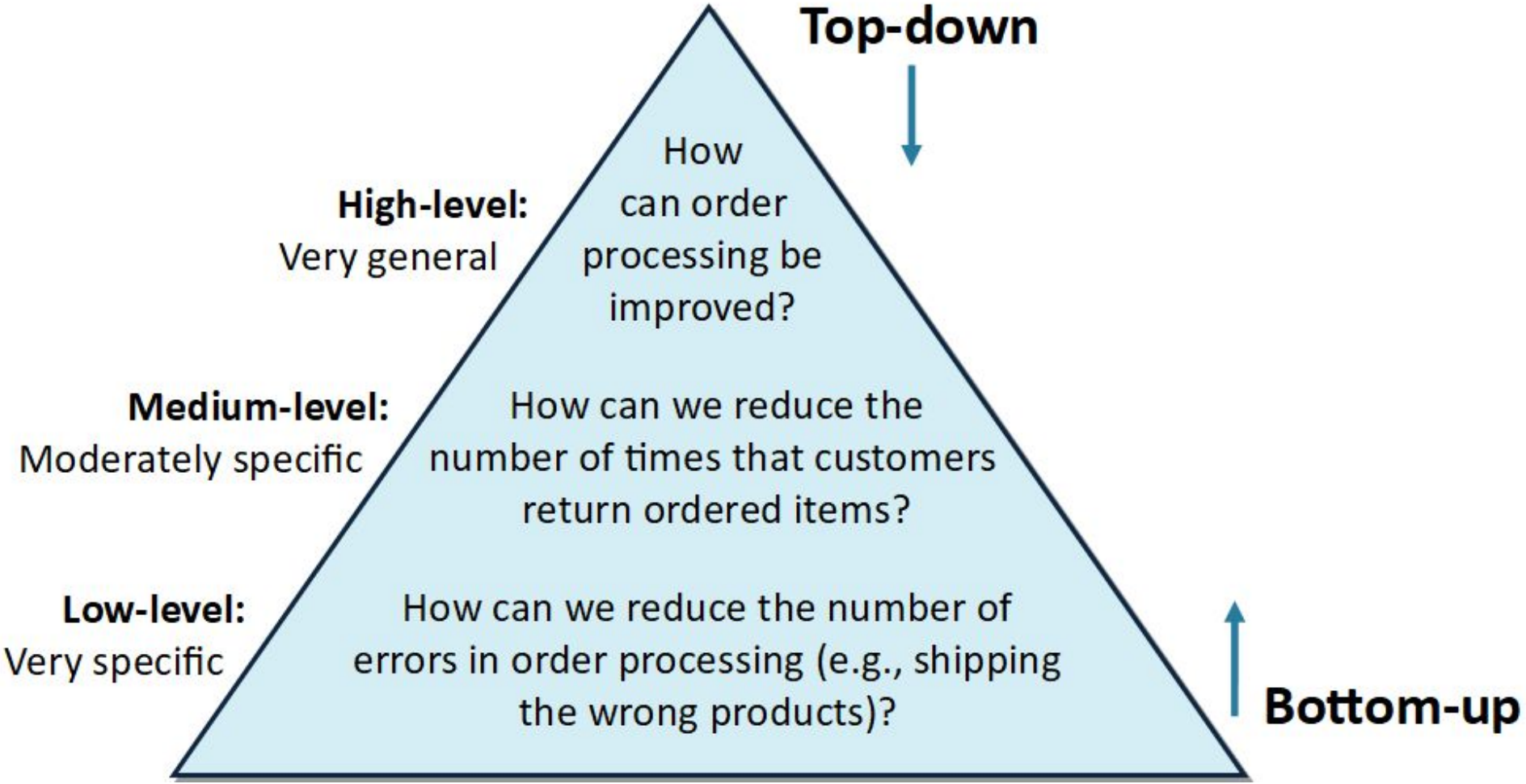
Most popular technique—if you need to know something, just ask

Process:

- Select people to interview & create a schedule
- Design interview questions (Open-ended, closed-ended, & probing types of questions)
- Prepare for the interview (Unstructured vs. structured interview organized in a logical order)
- Conduct the interview (Top-down vs. bottom-up)
- Follow-up after the interview

Question types

Types of Questions	Examples
Closed-ended questions	• How many telephone orders are received per day? • How do customers place orders? • What information is missing from the monthly sales report?
Open-ended questions	• What do you think about the current system? • What are some of the problems you face on a daily basis? • What are some of the improvements you would like to see in a new system?
Probing questions	• Why? • Can you give me an example? • Can you explain that in a bit more detail?


Top-down
High-level:
Very general
How can order processing be improved?
Medium-level:
Moderately specific
How can we reduce the number of times that customers return ordered items?
Low-level:
Very specific
How can we reduce the number of errors in order processing (e.g., shipping the wrong products)?
Bottom-up

Post interview

Interview Notes Approved by: Linda Estey

Person Interviewed: Linda Estey,

Director, Human Resources

Interviewer: Barbara Wixom

Purpose of Interview:

- Understand reports produced for Human Resources by the current system
- Determine information requirements for future system

Summary of Interview:

- Sample reports of all current HR reports are attached to this report. The information that is not used and missing information are noted on the reports.
- Two biggest problems with the current system are:
  1. The data are too old (the HR Department needs information within two days of month end; currently information is provided to them after a three-week delay)
  2. The data are of poor quality (often reports must be reconciled with departmental HR database)
- The most common data errors found in the current system include incorrect job level information and missing salary information.

Open Items:

- Get current employee roster report from Mary Skudrna (extension 4355).
- Verify calculations used to determine vacation time with Mary Skudrna.
- Schedule interview with Jim Wack (extension 2337) regarding the reasons for data quality problems.

Detailed Notes: See attached transcript.

Joint application development (JAD)

Joint user-analyst meeting hosted by a facilitator

- 10 to 20 users
- 1 to 2 scribes as needed to record the session
- Usually in a specially prepared room

Meetings can be held electronically and anonymously

- Reduces problems in group settings
- Can be held remotely
- Sessions require careful planning to be successful
- Users may need to bring documents or user manuals
- **Ground rules should be established**

Questionnaire

A set of written questions used to obtain information from individuals

May be paper based or electronic (e.g., web based)

Common uses:

- Large numbers of people
- Need both information and opinions
- When designing for use outside the organization (customers, vendors, etc.)
- Typical response rates: < 50% (paper); < 30% (Web)

Questionnaire steps

Select the participants

- Identify the population
- Use representative samples for large populations

Designing the questionnaire

- Careful question selection
- Remove ambiguities

Administering the questionnaire

- Working to get good response rate
- Offer an incentive (e.g., a free pen)

Questionnaire follow-up

- Send results to participants
- Send a thank-you

Good questionnaire design

- Begin with non-threatening and interesting questions
- Group items into logically coherent sections
- No important items at the very end
- Do not crowd a page with too many items
- Avoid abbreviations
- Avoid biased or suggestive items or terms
- Number questions to avoid confusion
- Pretest to identify confusing questions
- Provide anonymity to respondents

Document analysis

Provides information about the “as-is” system

Review technical documents when available

Review typical user documents:

- Forms
- Reports
- Policy manuals

Look for user additions to forms

Look for unused form elements

Observation

Users/managers often don’t remember everything they do

Checks validity of information gathered in other ways

Behaviors may change when people are watched
- Workers tend to be very careful when watched
- Keep a low profile
- Try not to interrupt or influence workers

Be careful not to ignore periodic activities
- Weekly … Monthly … Annually

Requirements-Gathering techniques compared

	Interviews	Joint Application Design	Questionnaires	Document Analysis	Observation
Type of information	As-is, improvements, to-be	As-is, improvements, to-be	As-is, improvements	As-is	As-is
Depth of information	High	High	Medium	Low	Low
Breadth of information	Low	Medium	High	High	Low
Integration of information	Low	High	Low	Low	Low
User involvement	Medium	High	Low	Low	Low
Cost	Medium	Low–Medium	Low	Low	Low to Medium

Other techniques

Concept Maps

- Represent meaningful relationships between concepts
- Focus individuals on a small number of key ideas

User Stories, Story Cards & Task Lists

- Associated with agile development methods
- Very low tech, high touch, easily updatable, and very portable
- Captured using story cards (index cards)
- Capture both functional and nonfunctional requirements

Concept Mapping

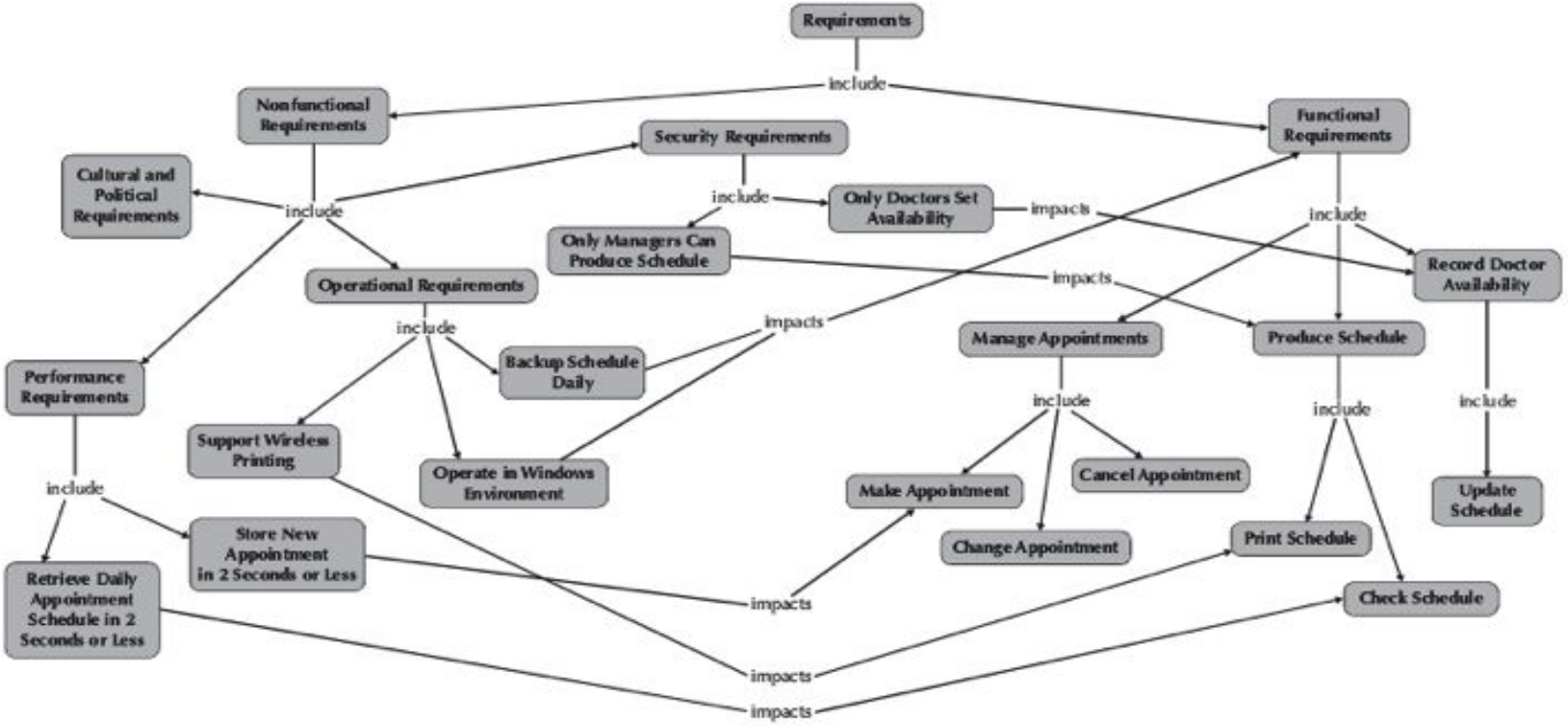


FIGURE 3-10 Sample Requirements Concept Map

Story cards / Task lists

Capture requirement using story cards (index cards)

File card with single requirement

Each requirement (card) is discussed

- How much effort is required to implement it
- A task list is created for each requirement (story)
- Large requirements can be split into smaller sections
- The story can be prioritized by risk level and importance

System proposal

Combines all material created in planning & analysis

Included sections:

- Executive summary
  - Provides all critical information is summary form
  - Helps busy executives determine which sections they need to read in more detail
- System request
- Workplan
- Feasibility analysis
- Requirements definition
- Current models of the system (expected to evolve)

Template

1. Table of Contents

2. Executive Summary

A summary of all the essential information in the proposal so a busy executive can read it quickly and decide what parts of the proposal to read in more depth.

3. System Request

The revised system request form (see Chapter 2).

4. Workplan

The original workplan, revised after having completed analysis (see Chapter 2).

5. Feasibility Analysis

A revised feasibility analysis, using the information from analysis (see Chapter 2).

6. Requirements Definition

A list of the functional and nonfunctional business requirements for the system (this chapter).

7. Functional Model

An activity diagram, a set of use case descriptions, and a use-case diagram that illustrate the basic processes or external functionality that the system needs to support (see Chapter 4).

8. Structural Models

A set of CRC cards, class diagram, and object diagrams that describe the structural aspects of the to-be system (see Chapter 5). This may also include structural models of the current as-is system that will be replaced.

9. Behavioral Models

A set of sequence diagrams, communication diagrams, behavioral-state machines, and a CRUDE matrix that describe the internal behavior of the to-be system (see Chapter 6). This may include behavioral models of the as-is system that will be replaced.

10. Appendices

These contain additional material relevant to the proposal, often used to support the recommended system. This might include results of a questionnaire survey or interviews, industry reports and statistics, and so on.

Brian Callahan, IS project manager, is just about ready to depart for an urgent meeting called by Joe Campbell, manager of manufacturing operations. A major project sponsored by Joe recently cleared the approval hurdle, and Brian helped bring the project through project initiation. Now that the approval committee has given the go-ahead, Brian has been working on the project's analysis plan. One evening, while playing golf with a friend who works in the manufacturing operations department, Brian learned that Joe wants to push the project's time frame up from Brian's original estimate of thirteen months. Brian's friend overheard Joe say, "I can't see why that IS project team needs to spend all that time analyzing things. They've got two weeks scheduled just to look at the existing system! That seems like a real waste. I want that team to get going on building my system." Because Brian has a little inside knowledge about Joe's agenda for this meeting, he has been considering how to handle Joe.

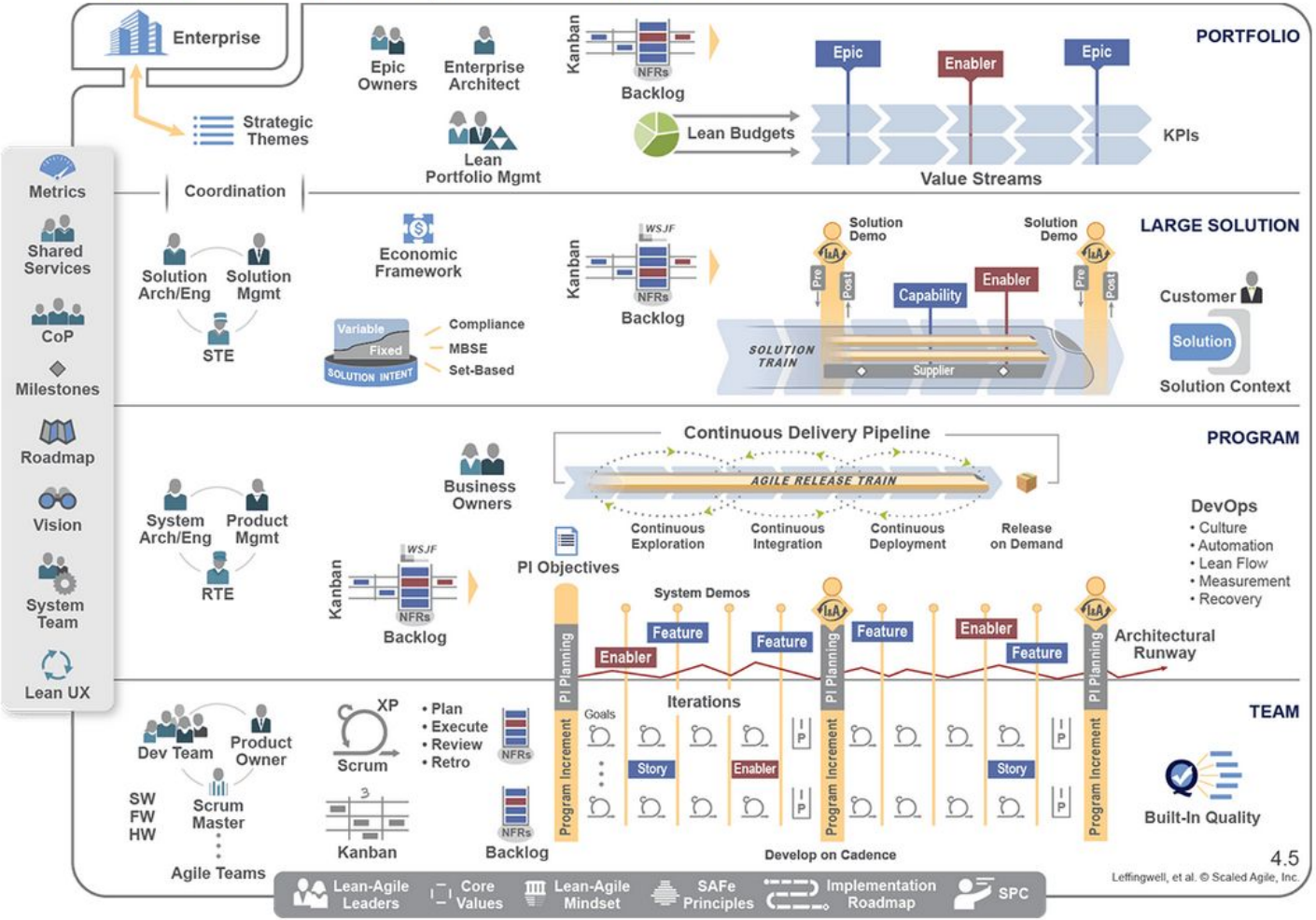
What do you suggest Brian tell Joe?

- Barry has recently been assigned to a project team that will be developing a new retail store management system for a chain of submarine sandwich shops. Barry has several years of experience in programming, but he has not done much analysis in his career. He was a little nervous about the new work he would be doing, but he was confident he could handle any assignment he was given. One of Barry's first assignments was to visit one of the submarine sandwich shops and prepare an observation report on how the store operates. Barry planned to arrive at the store around noon, but he chose a store in an area of town he was unfamiliar with, and due to traffic delays and difficulty in finding the store, he did not arrive until 1:30. The store manager was not expecting him and refused to let a stranger behind the counter until Barry had her contact the project sponsor (the director of store management) at company headquarters to verify who he was and what his purpose was. After finally securing permission to observe, Barry stationed himself prominently in the work area behind the counter so that he could see everything. The staff had to maneuver around him as they went about their tasks, but there were only minor occasional collisions. Barry noticed that the store staff seemed to be going about their work very slowly and deliberately, but he supposed that was because the store wasn't very busy. At first, Barry questioned each worker about what he or she was doing, but the store manager eventually asked him not to interrupt their work so much—he was interfering with their service to the customers. By 3:30, Barry was a little bored. He decided to leave, figuring he could get back to the office and prepare his report before 5:00 that day. He was sure his team leader would be pleased with his quick completion of his assignment. As he drove, he reflected, "There really won't be much to say in this report. All they do is take the order, make the sandwich, collect the payment, and hand over the order. It's really simple!" Barry's confidence in his analytical skills soared as he anticipated his team leader's praise. Back at the store, the store manager shook her head, commenting to her staff , "He comes here at the slowest time of day on the slowest day of the week. He never even looked at all the work I was doing in the back room while he was here—summarizing yesterday's sales, checking inventory on hand, making up resupply orders for the weekend . . . plus he never even considered our store-opening and -closing procedures. I hate to think that the new store management system is going to be built by someone like that. I'd better contact Chuck [the director of store management] and let him know what went on here today." 11
- **Evaluate Barry's conduct of the observation assignment.** 9

Does agile use requirements???

Well... it has to right? In some way...

SAFe® for Lean Enterprises — Full SAFe



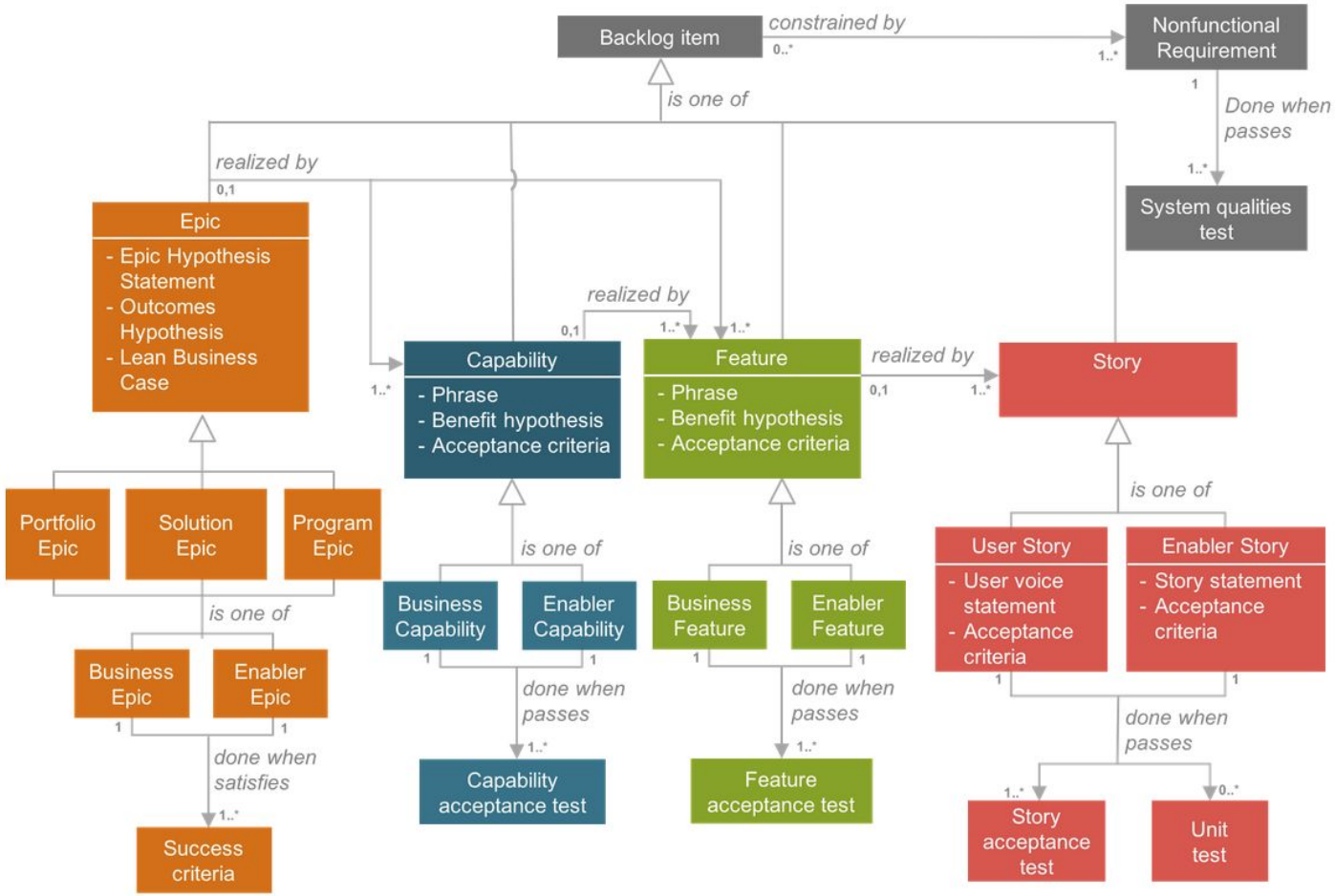
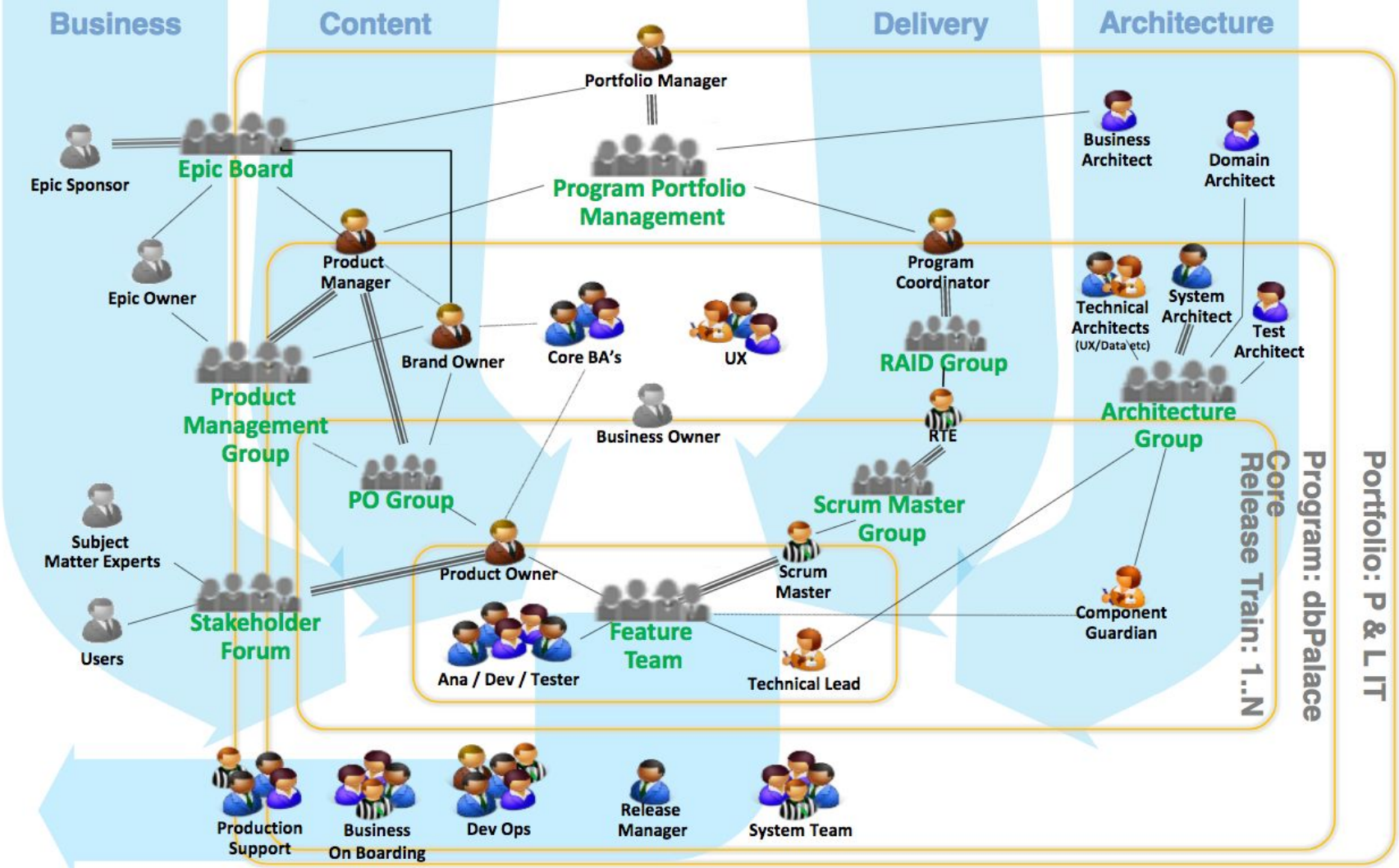
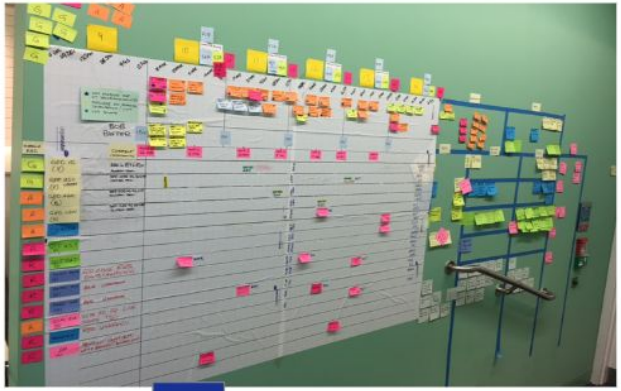


Figure 1. SAFe Requirements Model

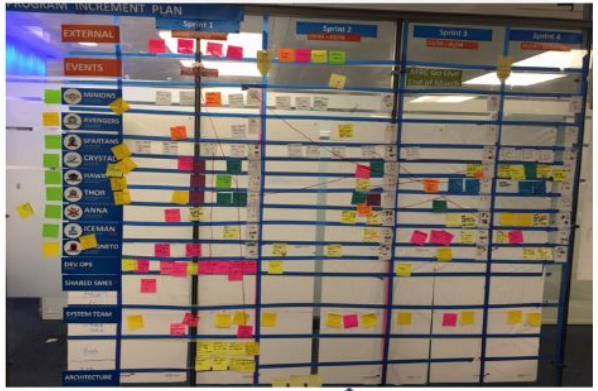

Business
Content
Delivery
Architecture
Portfolio Manager
Epic Sponsor
Epic Board
Program Portfolio Management
Business Architect
Domain Architect
Epic Owner
Product Manager
Program Coordinator
Technical Architects (UX/Data etc)
System Architect
Test Architect
Brand Owner
Core BA's
UX
RAID Group
Product Management Group
Business Owner
RTE
Architecture Group
PO Group
Scrum Master Group
Subject Matter Experts
Product Owner
Scrum Master
Users
Stakeholder Forum
Feature Team
Component Guardian
Ana / Dev / Tester
Technical Lead
Production Support
Business On Boarding
Dev Ops
Release Manager
System Team
Core Release Train: 1..N
Program: dbPalace
Portfolio: P & L IT

Planning and Backlog Grooming

EPIC Planning Board



PI Planning Board



- Long range epics and features detailed
- PI Velocity Published
- Relative Sizing applied
- Prioritised using WSJF
- Time critical events monitored
- Competing Epics Rationalised

Product Owners Board

Groomed Backlog

- Product Managers Leads the breakdown of the Epics & Features
- Business Design Ideas and optionality reviewed
- Maintain the theme of the EPIC Planning
- Manage committed velocity metrics to Epic Planning Velocity

As a project manager
I want to know if a story is running over
So that I can see if there is anything I can do to help.

168 Search by Name

As a help desk operator I want to search for my customers by their first and last names so that customer response times remain short

git demo

1. New project
2. Pull / Push
3. Pages