SENG 350
- Software Architecture & Design

Shuja Mughal

Why is Software Architecture important?

Fall 2024

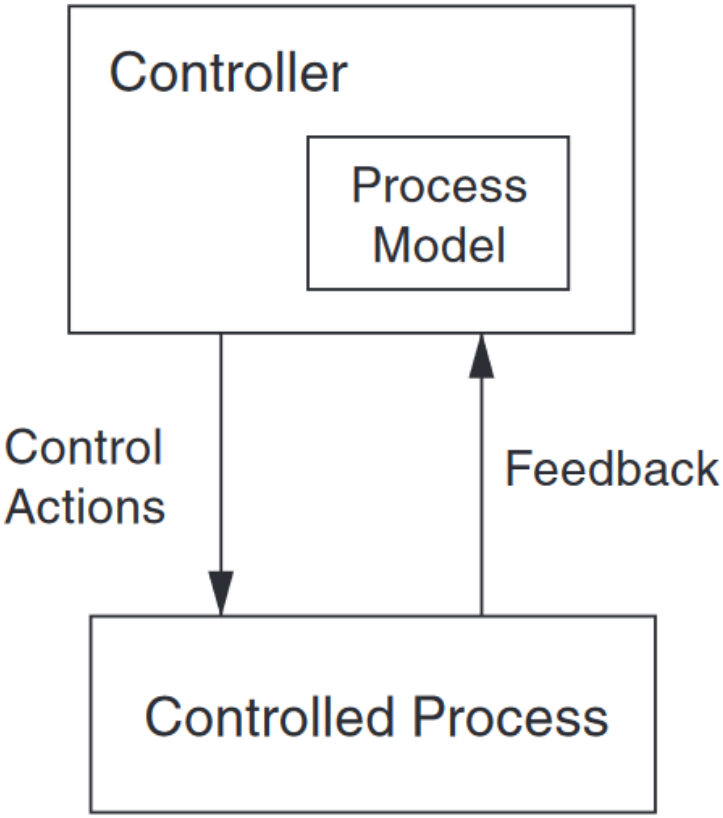
University of Victoria

Recall from last class

- SW Architecture is documented

University of Victoria

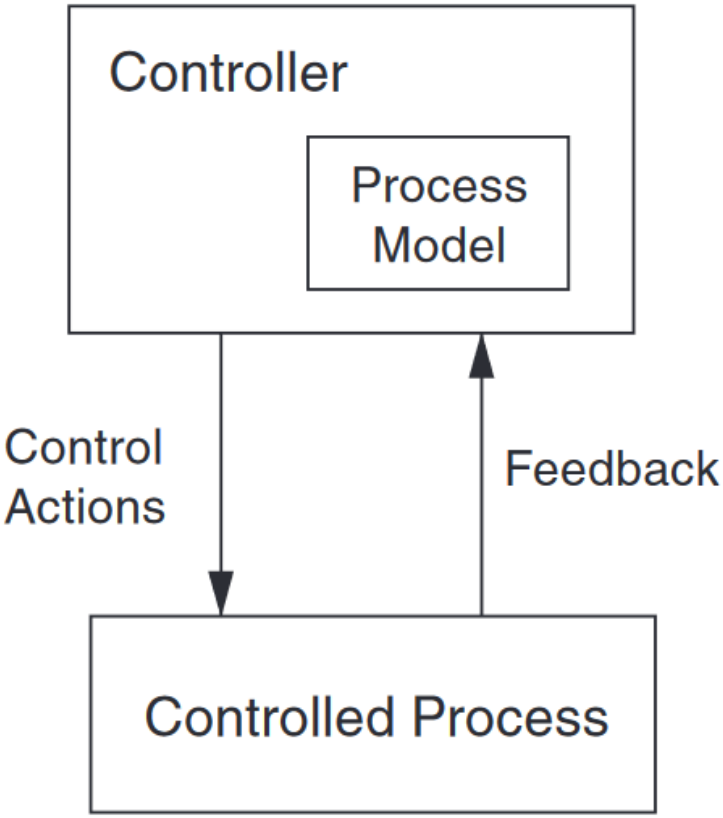
Which Views and how to relate?



System-Theoretic Accident Model and Processes (STAMP) by Leveson

University of Victoria

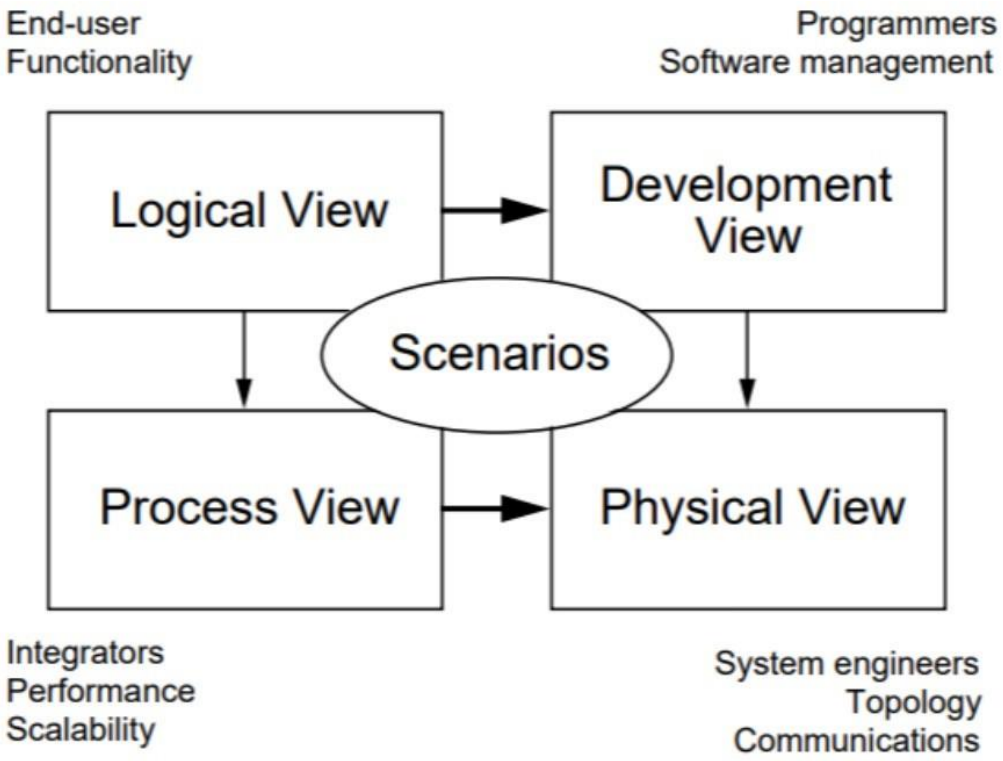
Which Views and how to relate?



System-Theoretic Accident Model and Processes (STAMP) by Leveson

University of Victoria

Kruchten's 4+1 View Model



https://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf

University of Victoria

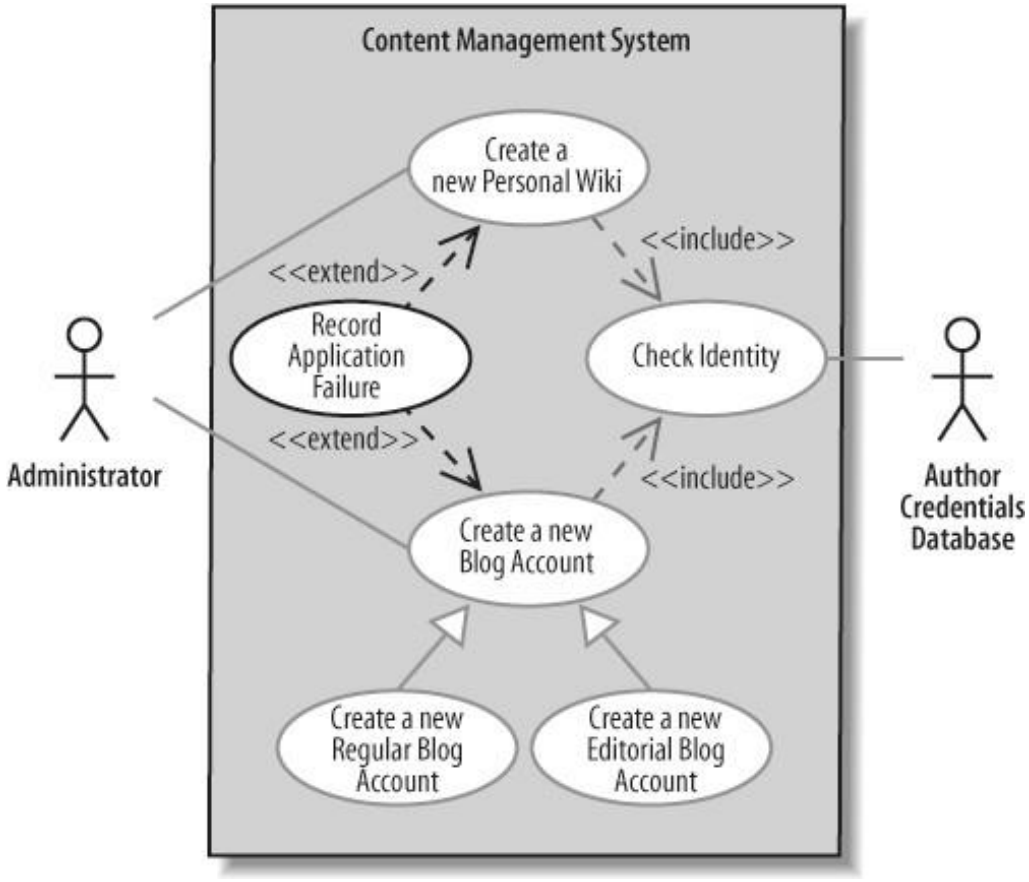
Scenarios are in the realm of Requirements Engineering

How to describe?

University of Victoria

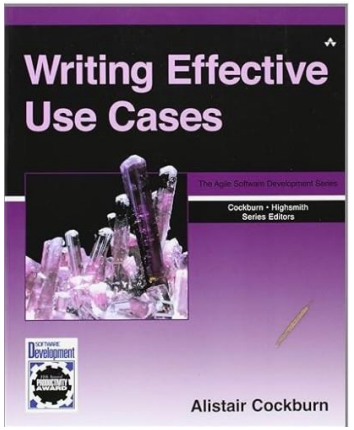
Jens H. Weber

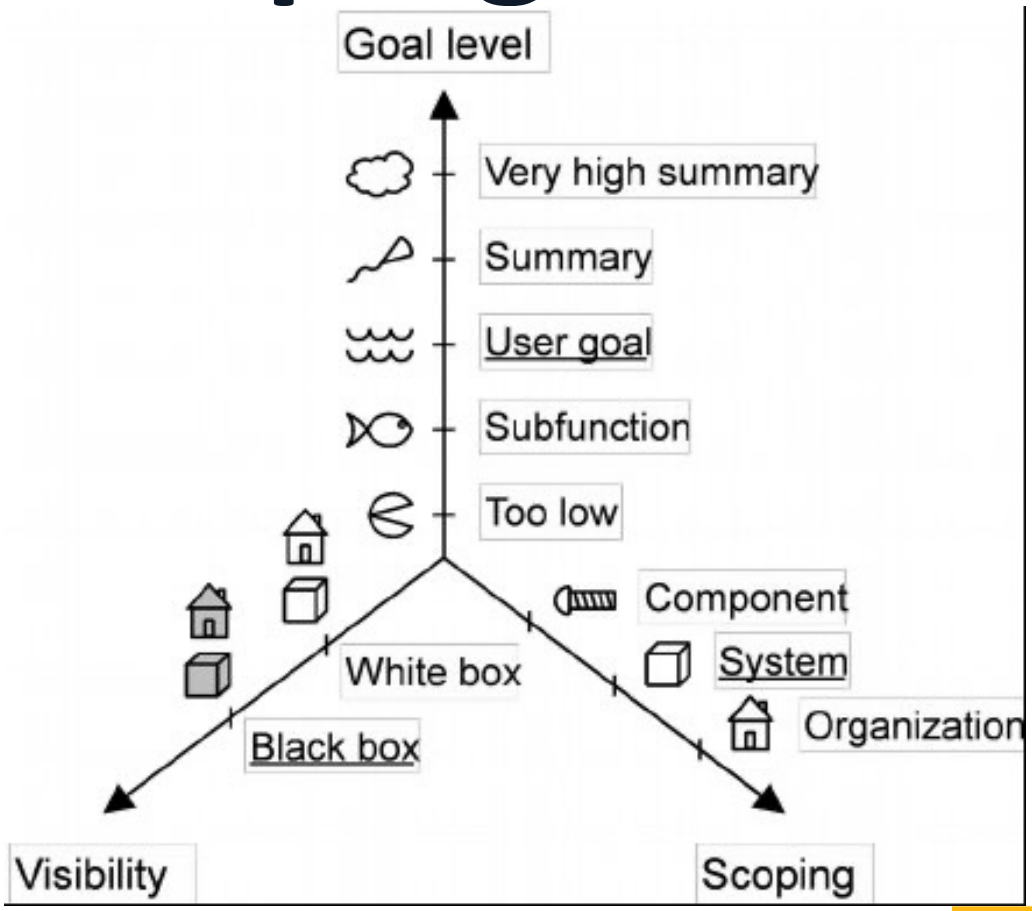
Use Case Modelling



University of Victoria

Scoping of Use Cases





University of Victoria

Textual Specification

USE CASE #	< the name is the goal as a short active verb phrase>	
Goal in Context	<a longer statement of the goal in context if needed>	
Scope & Level	<what system is being considered black box under design> <one of : Summary, Primary Task, Subfunction>	
Preconditions	<what we expect is already the state of the world>	
Success End Condition	<the state of the world upon successful completion>	
Failed End Condition	<the state of the world if goal abandoned>	
Primary, Secondary Actors	<a role name or description for the primary actor>. <other systems relied upon to accomplish use case>	
Trigger	<the action upon the system that starts the use case>	
DESCRIPTION	**Step**	**Action**
	1	<put here the steps of the scenario from trigger to goal delivery,and any cleanup afte>
	2	<...>
	3	
EXTENSIONS	**Step**	**Branching Action**
	1a	<condition causing branching> : <action or name of sub.use case>
SUB-VARIATIONS		**Branching Action**
	1	<list of variation s>

University of Victoria

Textual Specification (related info)

RELATED INFORMATION	<Use case name>
Priority:	<how critical to your system / organization>
Performance	<the amount of time this use case should take>
Frequency	<how often it is expected to happen>
Channels to actors	<e.g. interactive, static files, database, timeouts>
OPEN ISSUES	<list of issues awaiting decision affecting this use case >
Due Date	<date or release needed>
...any other management information...	<...as needed>
Superordinates	<optional, name of use case(s) that includes this one>
Subordinates	<optional, depending on tools, links to sub.use cases>

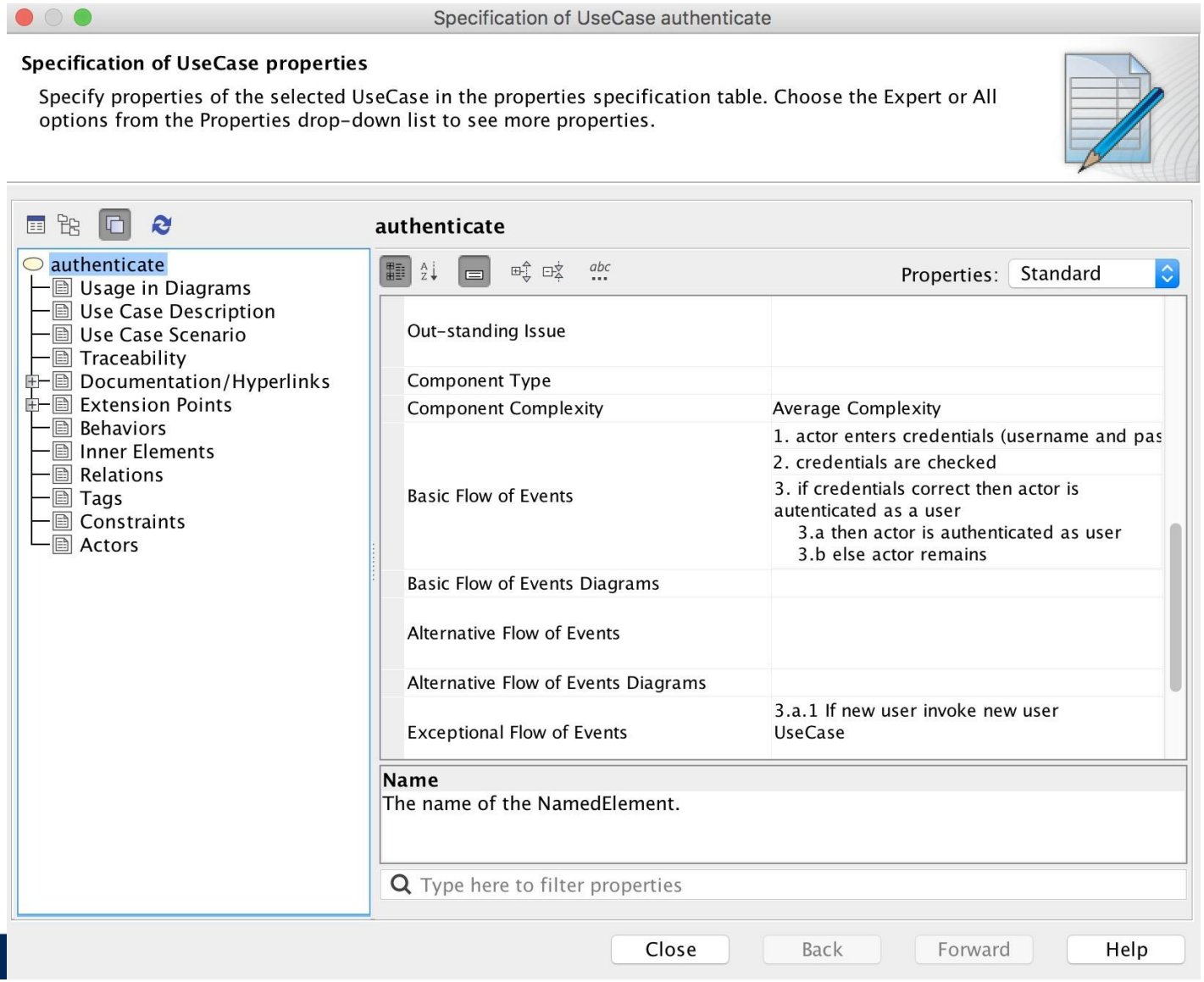
USE CASE 5	Buy Goods
Goal in Context	Buyer issues request directly to the company, expects goods shipped and to be billed.
Scope & Level	Company, Summary
Preconditions	We know Buyer, their address, etc.
Success End Condition	The buyer has goods; we have money for the goods.
Failed End Condition	We have not sent the goods, Buyer has not spent the money.
Primary, Secondary Actors	Buyer, any agent (or computer) acting for the customer. Credit card company, bank, shipping service

DESCRIPTION	Step	Action
	1	Buyer calls in with a purchase request
	2	Company captures buyer's name, address, requested goods, etc.
	3	Company gives buyer information on goods, prices, delivery dates, etc.
	4	Buyer signs for order.
	S	Company creates order, ships order to buyer.
	6	Company ships invoice to buyer.
	7	Buyers pays invoice.
EXTENSIONS	Step	Branching Action
	3a	Company is out of one of the ordered items: 3a1. Renegotiate order.
	4a	Buyer pays directly with credit card: 4a1. Take payment by credit card (use case 44)
	7a	Buyer returns goods: 7a. Handle returned goods (use case 105)

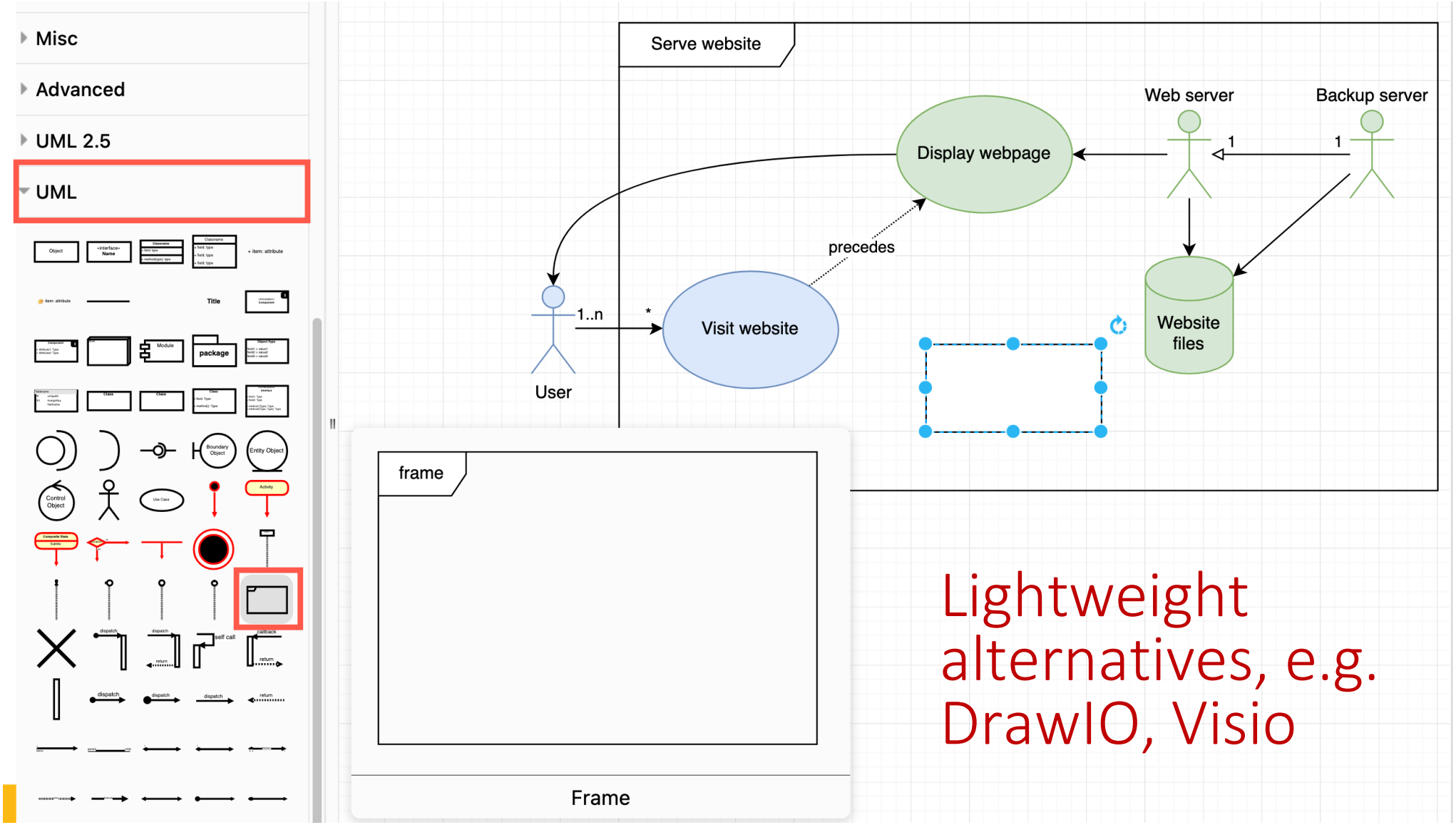
EXTENSIONS	**Step**	**Branching Action**
	3a	Company is out of one of the ordered items: 3a1. Renegotiate order.
	4a	Buyer pays directly with credit card: 4a1. Take payment by credit card (use case 44)
	7a	Buyer returns goods: 7a. Handle returned goods (use case 105)
SUB-**VARIATIONS**		Branching Action
	1	Buyer may use phone in, fax in, use web order form, electronic interchange
	7	Buyer may pay by cash or money order check credit card

RELATED INFORMATION	5. Buy Goods
Priority:	t P
Performance	5 minutes for order, 45 daya until paid
Frequency	200/day
Channel to actors	not yet determined
OPEN ISSUES	What if we have pa8 of the order? What is credit card is stolen?
Due Date	release 1.0
...any other management information...	
Superordinates	Manage customer relationship (use caae 2)
Subordinates	Create order (use case 15) Take payment by credit card (uxe case 44)

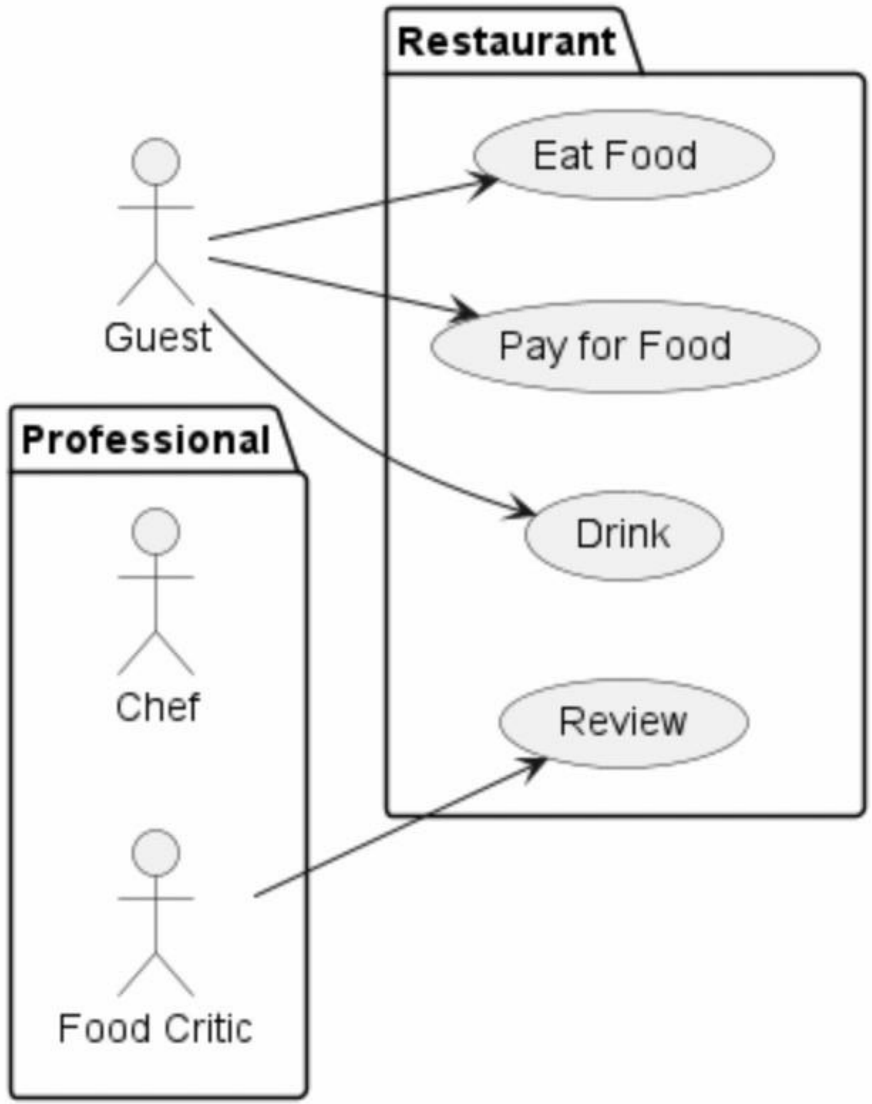
CASE Tools provide integrated modelling



University of Victoria


Misc
Advanced
UML 2.5
UML
Serve website
Web server
Backup server
Display webpage
1
1
precedes
1..n
*
Visit website
User
Website files
frame
Frame
Lightweight alternatives, e.g. DrawIO, Visio

```
@startuml
left to right direction
actor Guest as g
package Professional {
  actor Chef as c
  actor "Food Critic" as fc
}
package Restaurant {
  usecase "Eat Food" as UC1
  usecase "Pay for Food" as UC2
  usecase "Drink" as UC3
  usecase "Review" as UC4
}
fc --> UC4
g --> UC1
g --> UC2
g --> UC3
@enduml
```



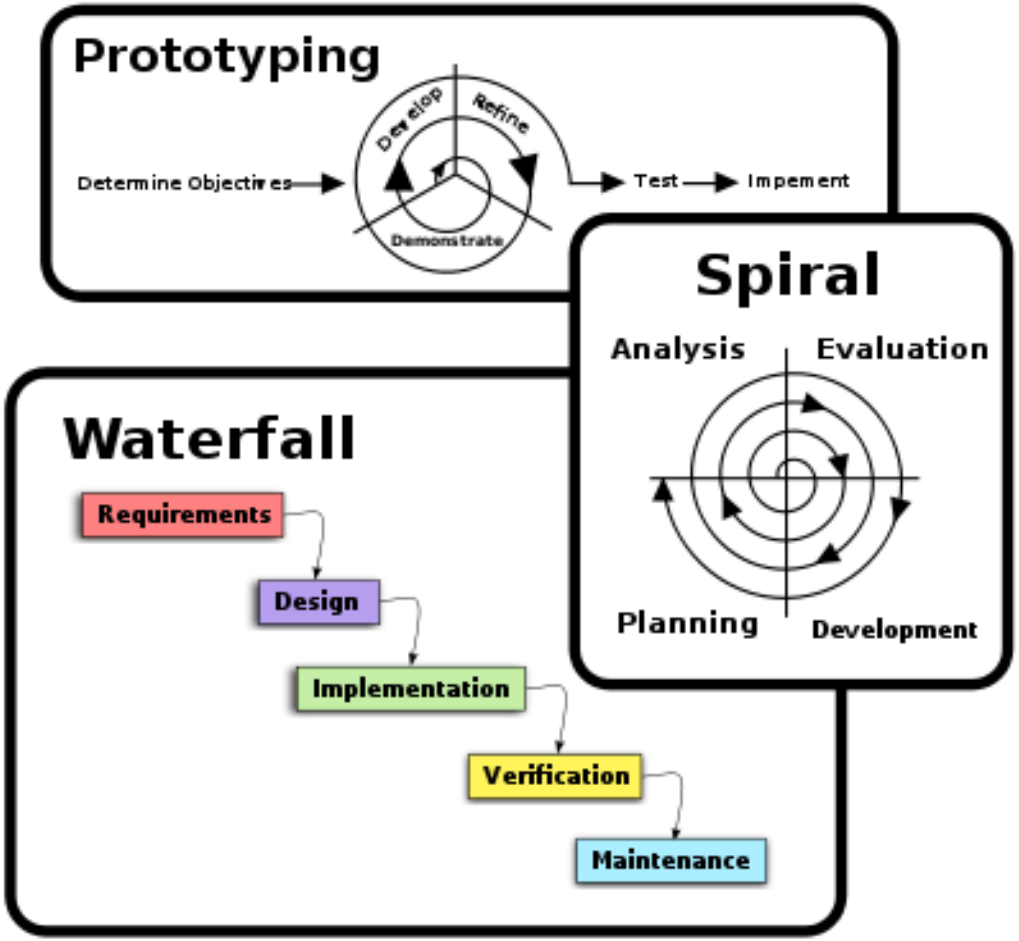
University of Victoria

Textual alternatives, e.g. Plantuml, Mermaid

Project Milestone 1 - Requirements

Medical Information System To Enhance Resources for Emergency Departments - Mister Ed
Emergency Departments (EDs) are often crowded and overloaded. The Mister Ed system will be designed to help with this situation. People who feel that they need to visit an ED will be able to use Mister Ed to understand the current load of EDs in their area. They will be able to register virtually and undergo a "virtual triage" to determine whether they really should visit the ER or potentially follow another course of action, like going to a regular primary care clinic (GP), taking over-the-counter medication, or contact the nurse/clinician hotline over the phone or Internet. Patients who can safely be triaged virtually but still need to visit the ED can wait from the comfort of their home and will be notified to come in when it is time to see them. Patients who need to be triaged in person are asked to attend but can then return to their homes until it is time to see them.

In groups of 2, use https://plantuml.com to create a UseCase Diagram for MisterEd. Submit to Teams Week 2. Take ~20 Minutes



Shuja Mughal
University of Victoria

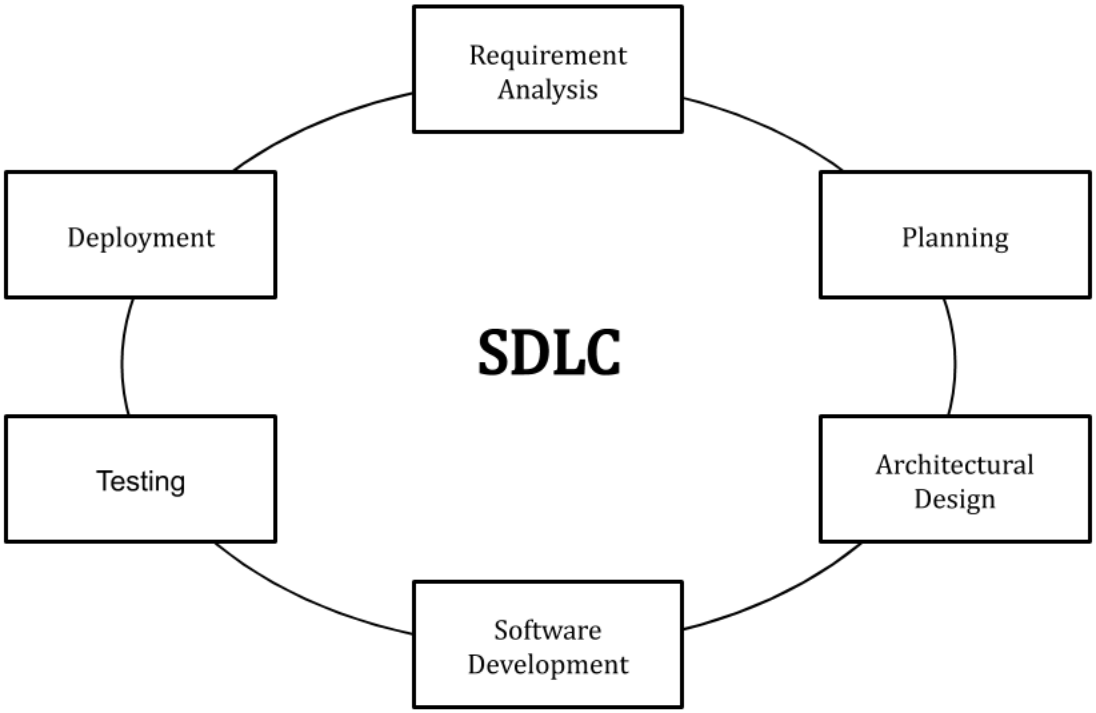
SW Process Models

University of Victoria

What is a Software Process Model?

In software engineering, a **software development methodology** (also known as a **system development methodology**, **software development life cycle**, **software development process**, **software process**) is a splitting of software development work into distinct phases (or stages) containing activities with the intent of better planning and management. It is often considered a subset of the systems development life cycle. The methodology may include the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application.

Software Development Life Cycle



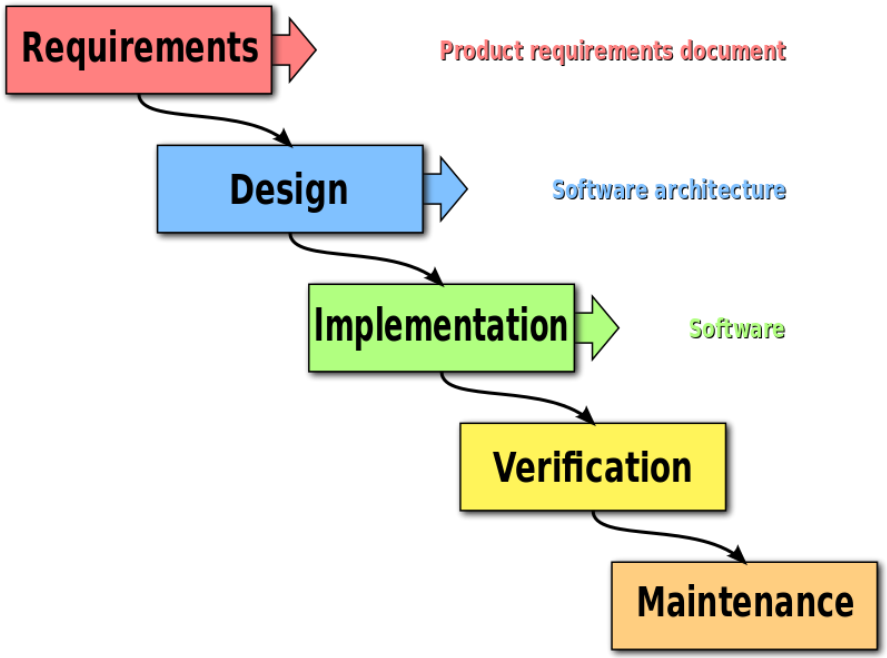
The Software Development Life Cycle (SDLC) is a structured process that enables the production of high-quality, low-cost software, in the shortest possible production time.
The SDLC's goal is to produce high-quality software that meets and exceeds all customer expectations and demands.
The SDLC defines and outlines a detailed plan with stages, or phases, each encompassing its own process and deliverables.

University of Victoria

Benefits of SDLC

- Lower development costs.
- Improved quality of software products.
- More visibility over the activities of the development team.
- Quicker time-to-market due to better organization, more transparency, and fewer after-the-fact fixes.
- More precise project planning, budget estimations, and scheduling.
- Improved communication between different teams and upper management.
- Less chance of project failure.
- Improved quality and exactness of documentation.
- An in-depth understanding of customer and business needs.
- A team culture that emphasizes knowledge sharing and continuous learning.

The Waterfall Model



The project is divided into sequential phases, with some overlap and splashback acceptable between phases.
The emphasis is on planning, time schedules, target dates, budgets, and the implementation of an entire system at once.
Tight control is maintained over the project's life via extensive written documentation, formal reviews, and approval/signoff by the user and information technology management at the end of most phases before beginning the next phase. Written documentation is an explicit deliverable of each phase.

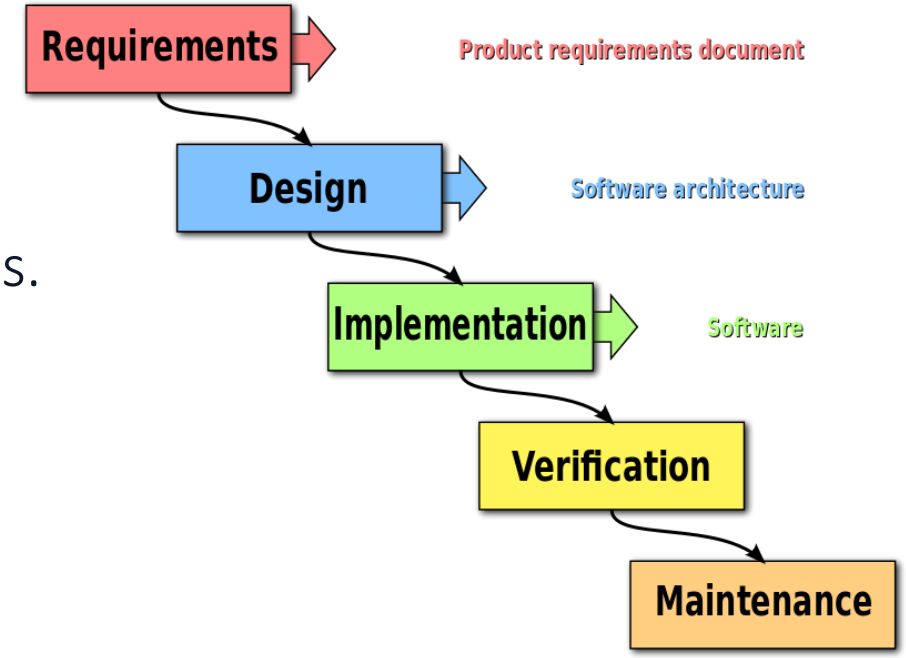
[Royce, 70's]

University of Victoria

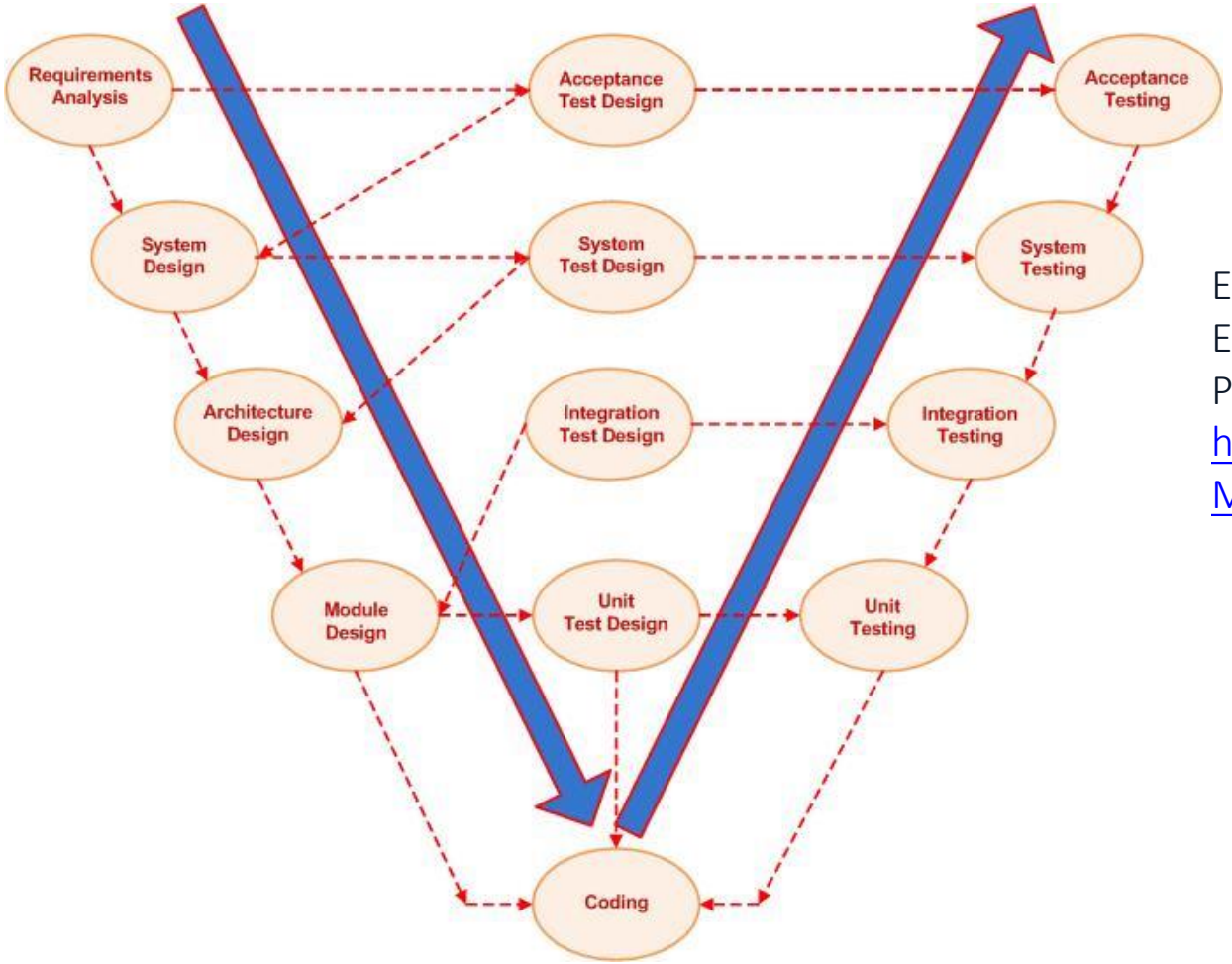
The Waterfall Model

When to use the Waterfall Model?

- For small projects
- When requirements are clear
- When customer involvement is less.
- For low-budget projects.



University of Victoria



The V Model

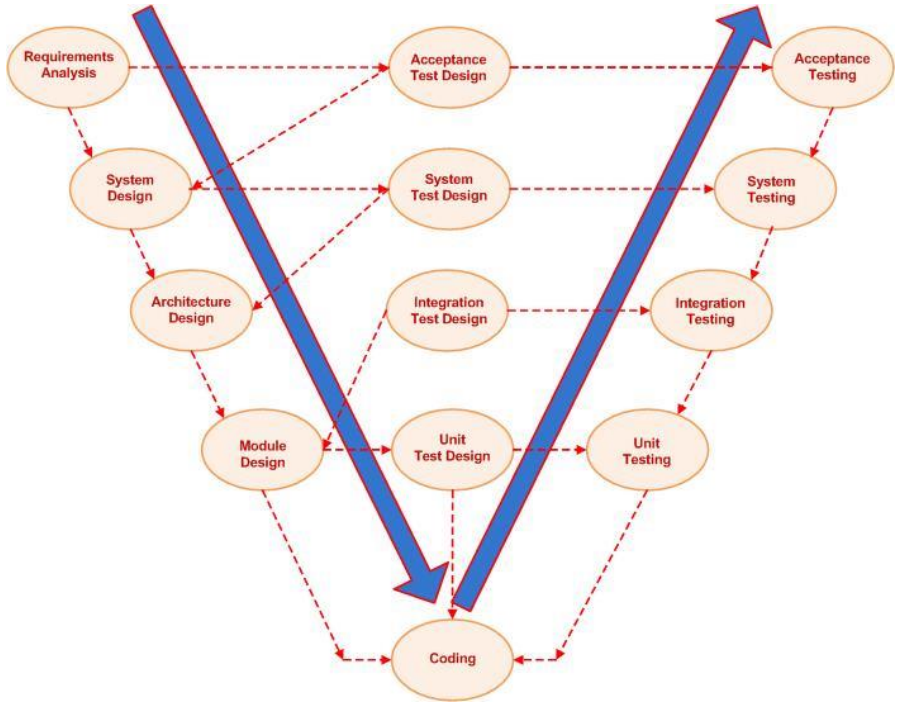
Extension of the Waterfall model
Emphasis of Verification and Validation (V&V)
Popular in safety-critical systems
https://en.wikipedia.org/wiki/V-Model_(software_development)

University of Victoria

The V Model

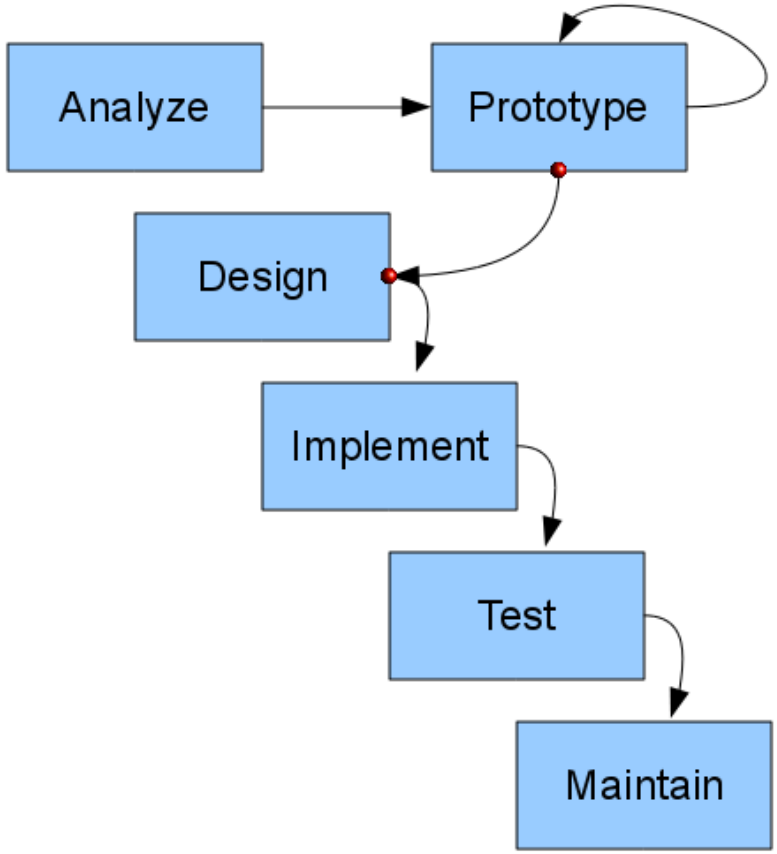
When to use the V Model?

- When the requirement is well-defined and not ambiguous.
- The V-shaped model should be used for projects where requirements are clearly defined and fixed.
- The V-shaped model should be chosen when sample technical resources with essential technical expertise are available.



University of Victoria

The Prototyping Model



Prototyping is not a standalone, complete development methodology, but rather an approach to try out particular features in the context of a full methodology (such as incremental, spiral, or rapid application development (RAD)).

Attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.

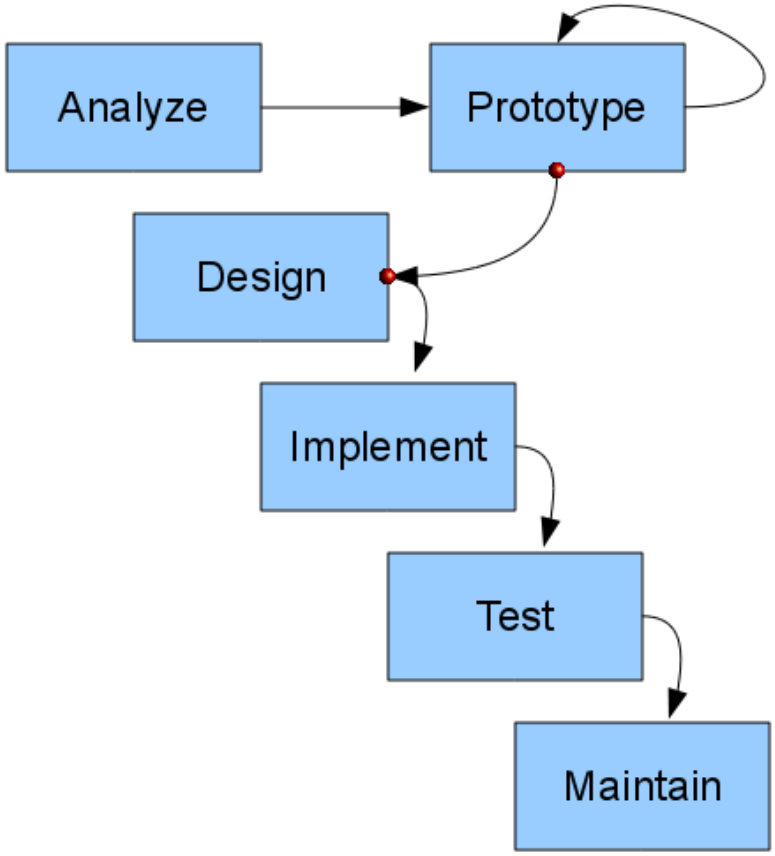
The client is involved throughout the development process, which increases the likelihood of client acceptance of the final implementation.

While some prototypes are developed with the expectation that they will be discarded, it is possible in some cases to evolve from prototype to working system.

The Prototyping Model

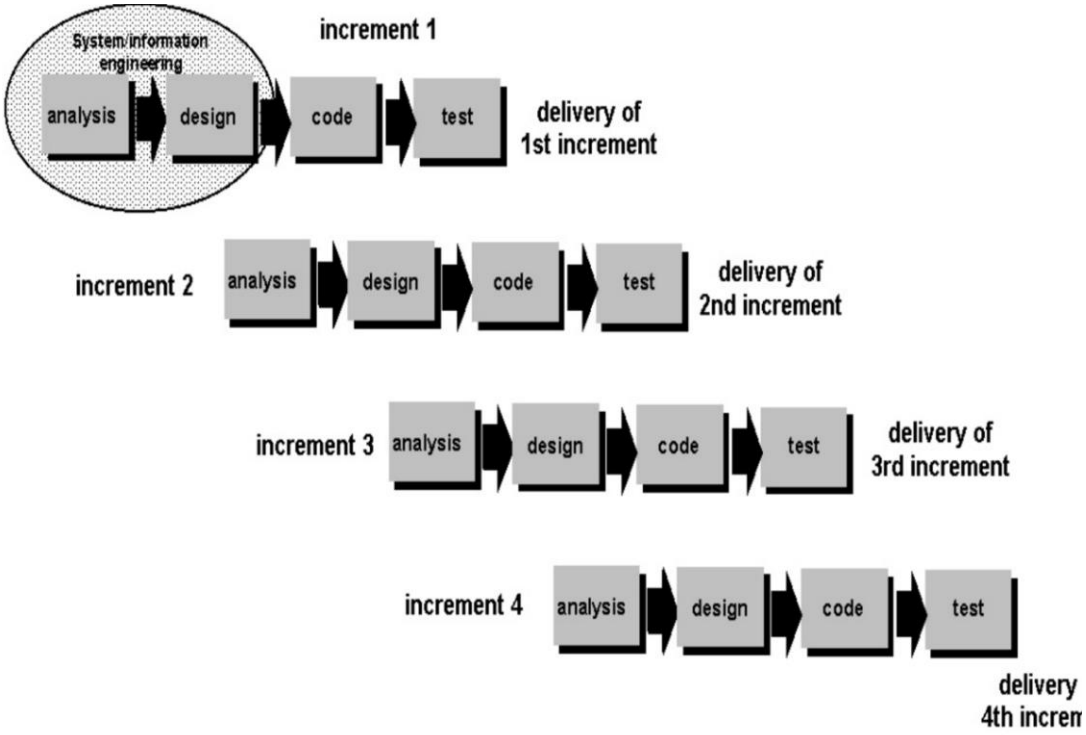
When to use the Prototyping Model?

- The prototype model should be use when the product's requirements ar unclear or unstable.
- It can also be used if requirements ar changing quickly.
- This model can successfully be used t develop user interfaces, high-tec software-intensive systems, and system with complex algorithms and interfaces.
- It is also a perfect choice to demonstrate the product’s technical feasibility.



University of Victoria

The Incremental Model



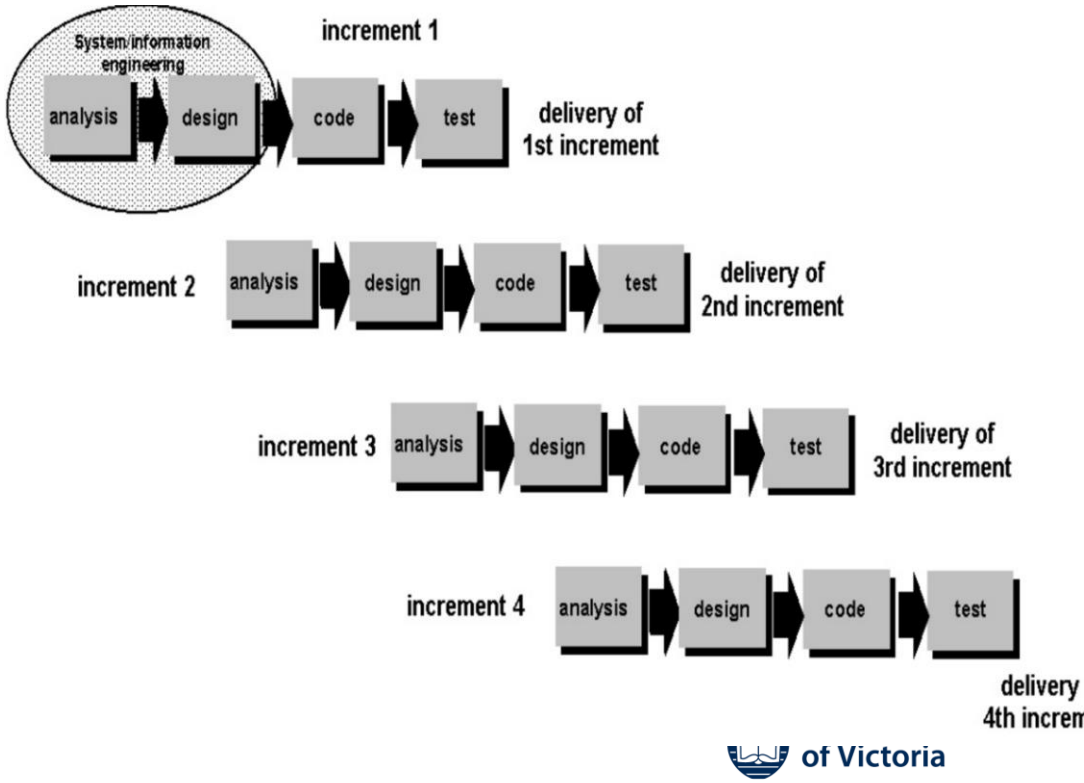
A series of mini-Waterfalls are performed, where all phases of the Waterfall are completed for a small part of a system, before proceeding to the next increment, or
Overall requirements are defined before proceeding to evolutionary, mini-Waterfall development of individual increments of a system, or
The initial software concept, requirements analysis, and design of architecture and system core are defined via Waterfall, followed by incremental implementation, which culminates in installing the final version, a working system.

University of Victoria

The Incremental Model

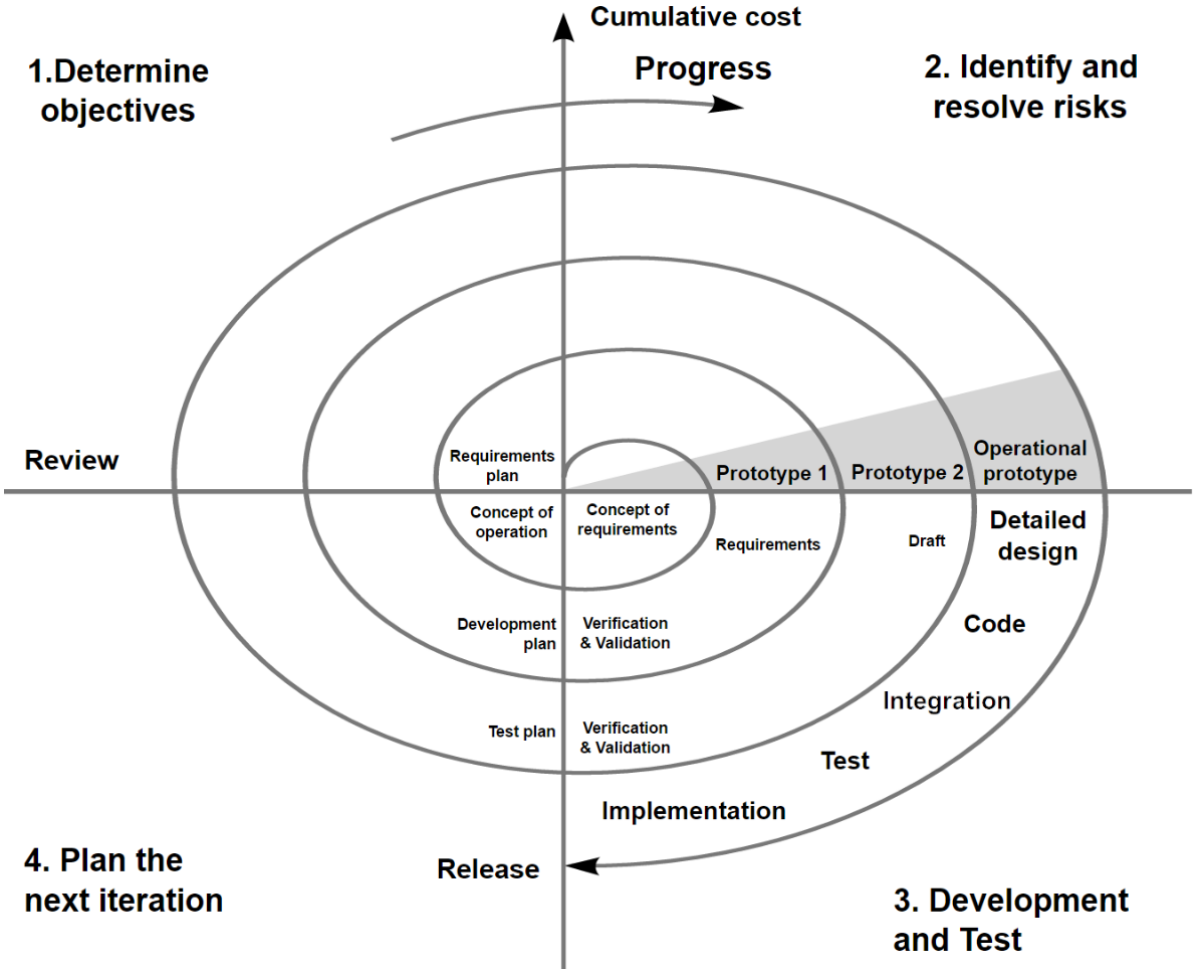
When to use the Incremental Model?

- When major requirements are understood but some requirements can evolve within the passage of time.
- When product launch in the market is getting late.
- When a customer has no problem with the budget, but he demands more and more quality in software.



The Spiral Model

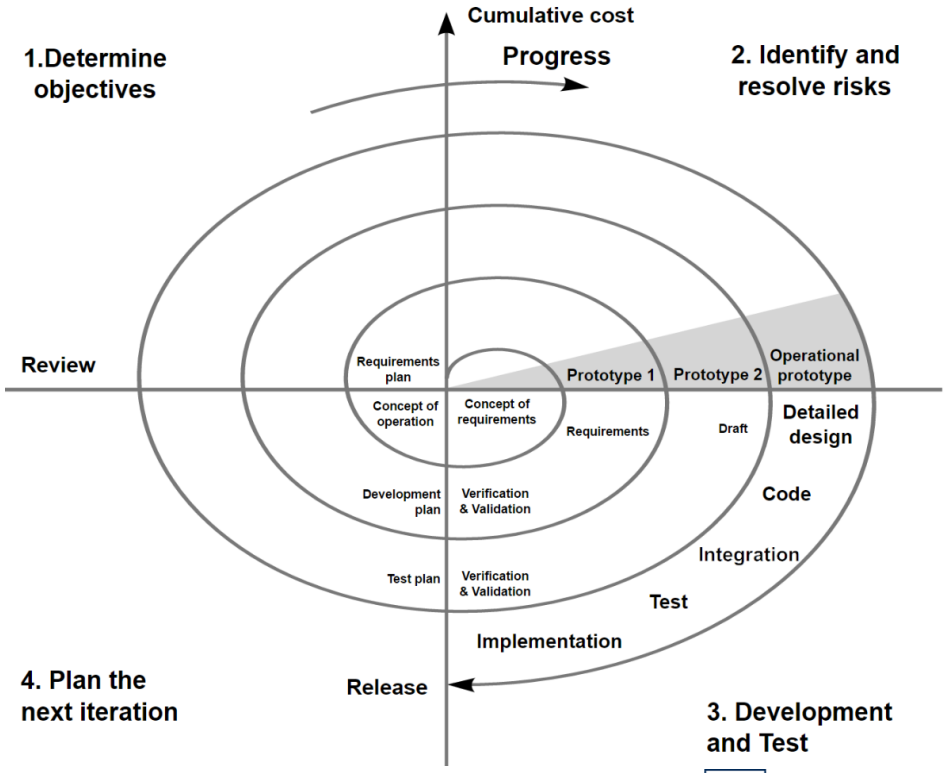
The focus is on risk assessment, breaking a project into smaller segments, and providing more ease of change during the development process.
Each cycle is a progression through the same sequence of steps, for each part of the product and for each of its levels of elaboration. This iterative process, from an overall concept-of-operation document down to the coding of each program, is a key aspect of our development strategy.
Begin each cycle by identifying stakeholders and their "win conditions," and end each cycle with review and commitment.



The Spiral Model

When to use the Spiral Model?

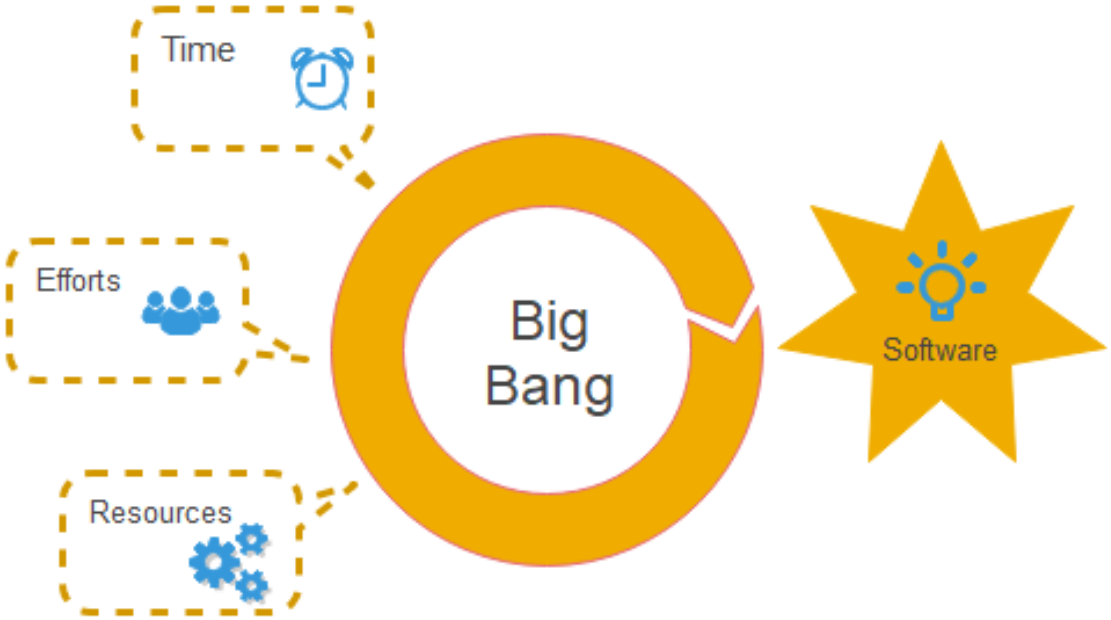
- When the risk is medium or high
- For real-time systems
- For large projects
- When requirements are not clear
- When changes in the software can be expected at any time.



University of Victoria

The Big Bang Model

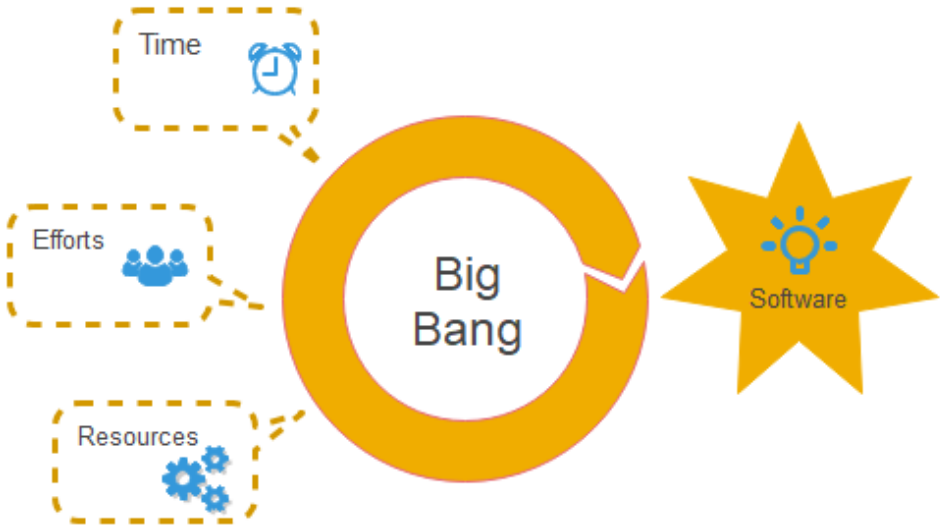
In this model, developers do not follow any specific process. Development begins with the necessary funds and efforts in the form of inputs.
The result may or may not be as per the customer's requirement, because in this model, even the customer requirements are not defined.



The Big Bang Model

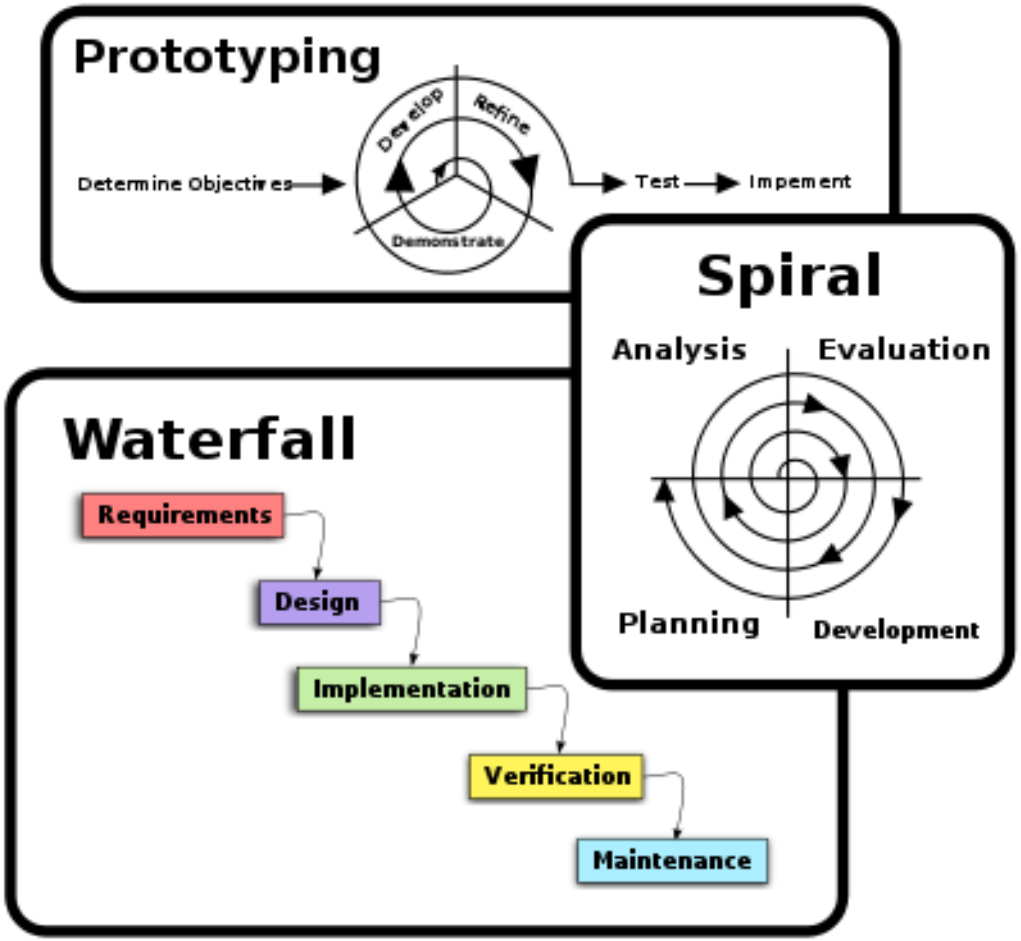
When to use the Big Bang Model?

- This model is required when this project is small, like an academic or practical project.
- This method is also used when the developer team is small, requirements are not defined, and the release date is not confirmed or given by the customer.



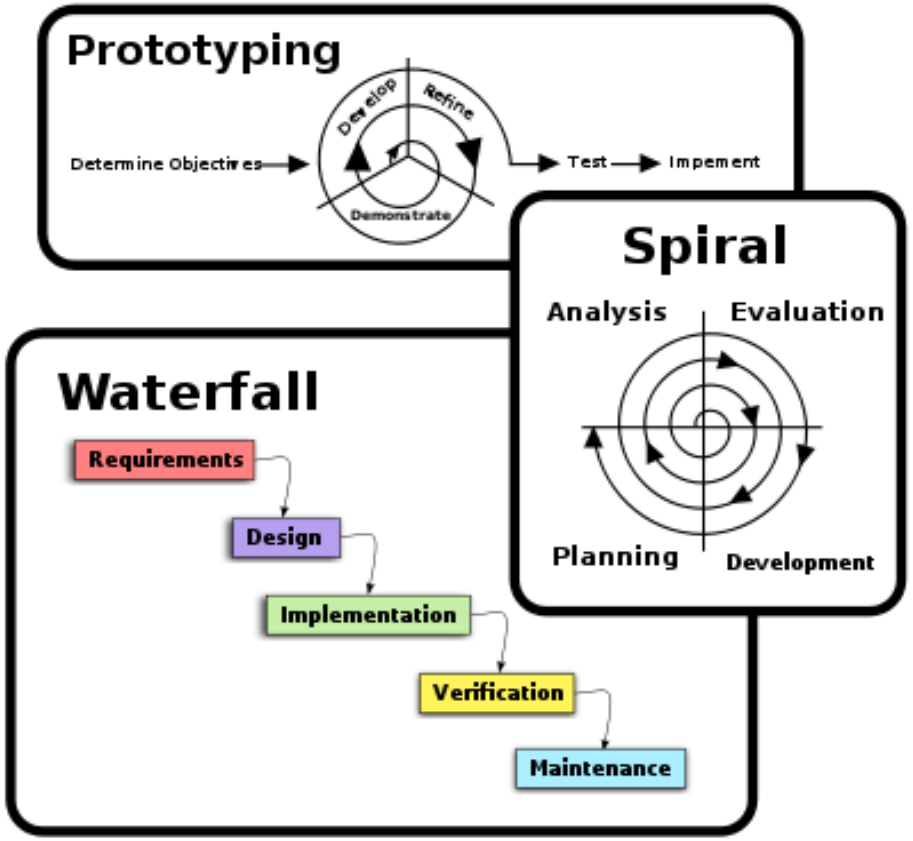
University of Victoria

Let’s look at some Case studies



University of Victoria

Case Study 1

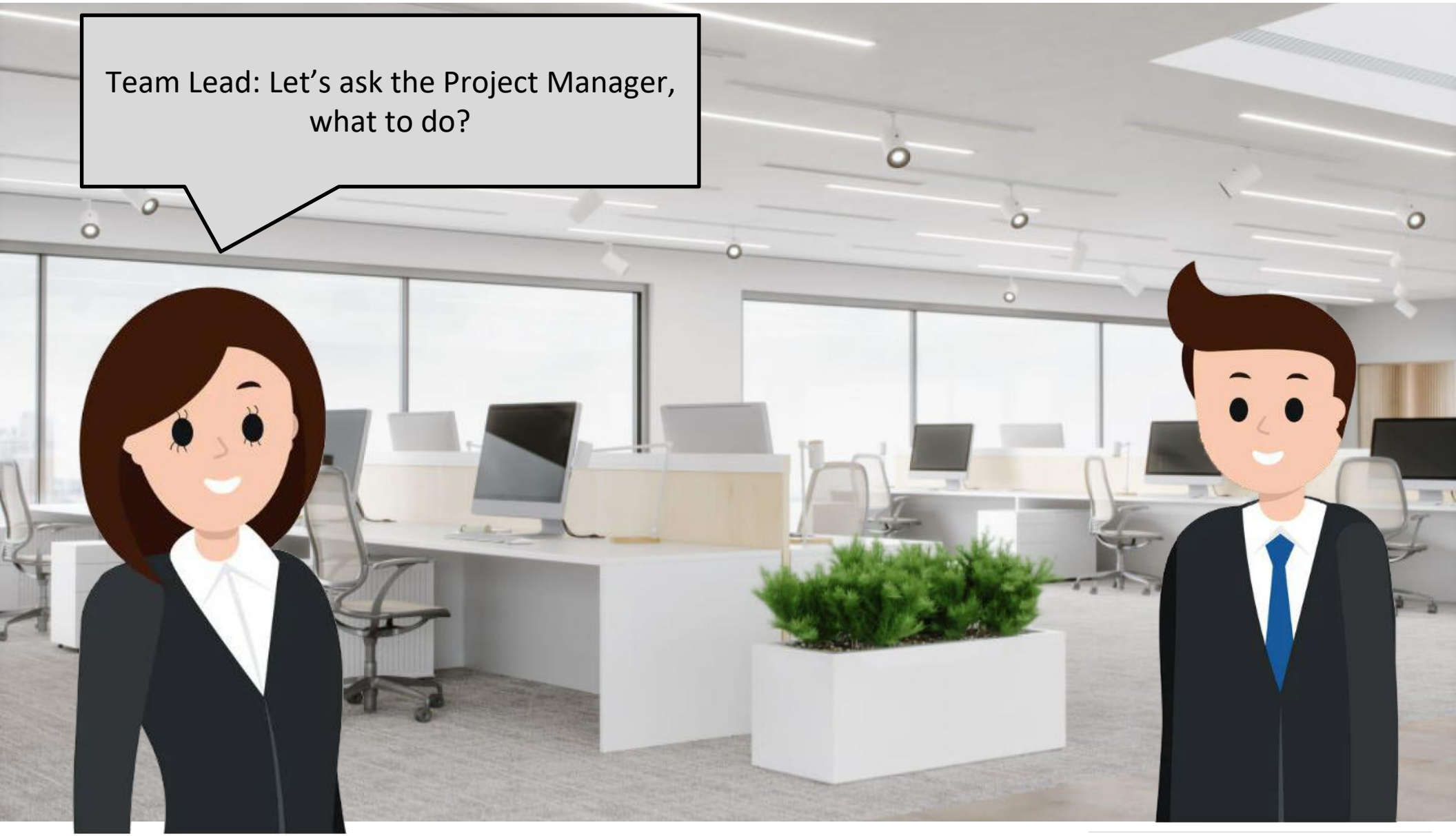


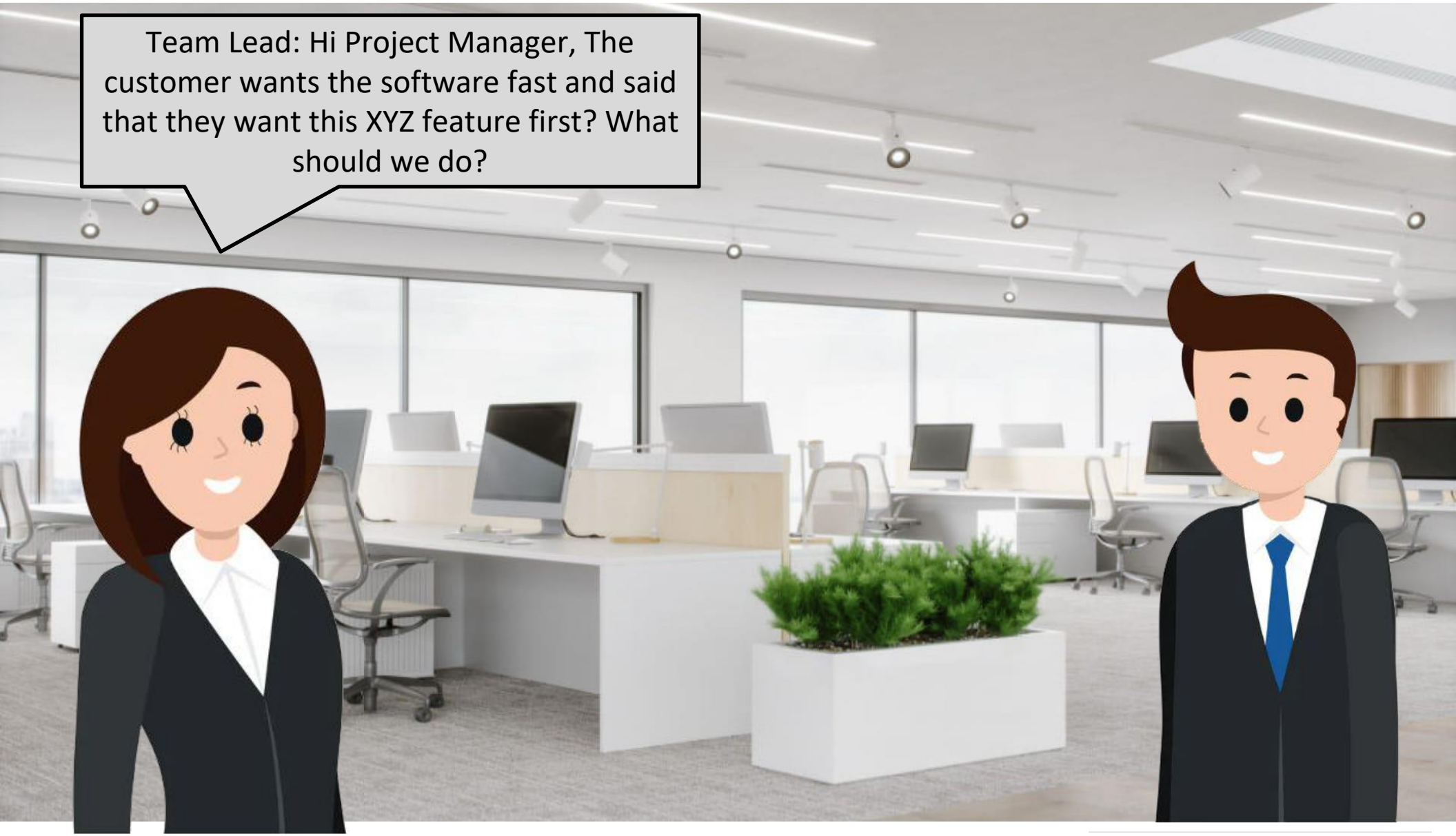
University of Victoria

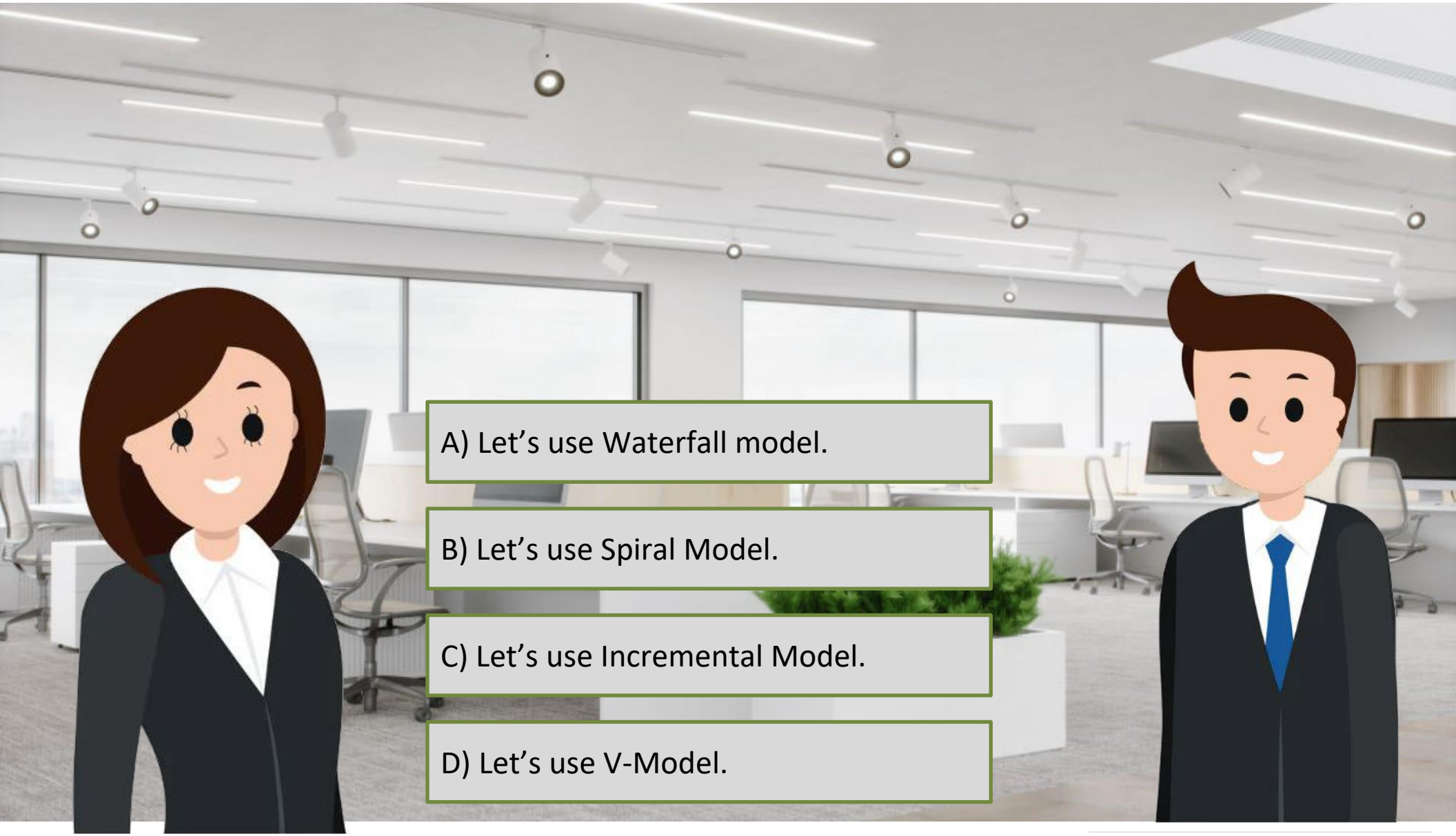

Employee: Team lead, the client wants their company’s business software fast?

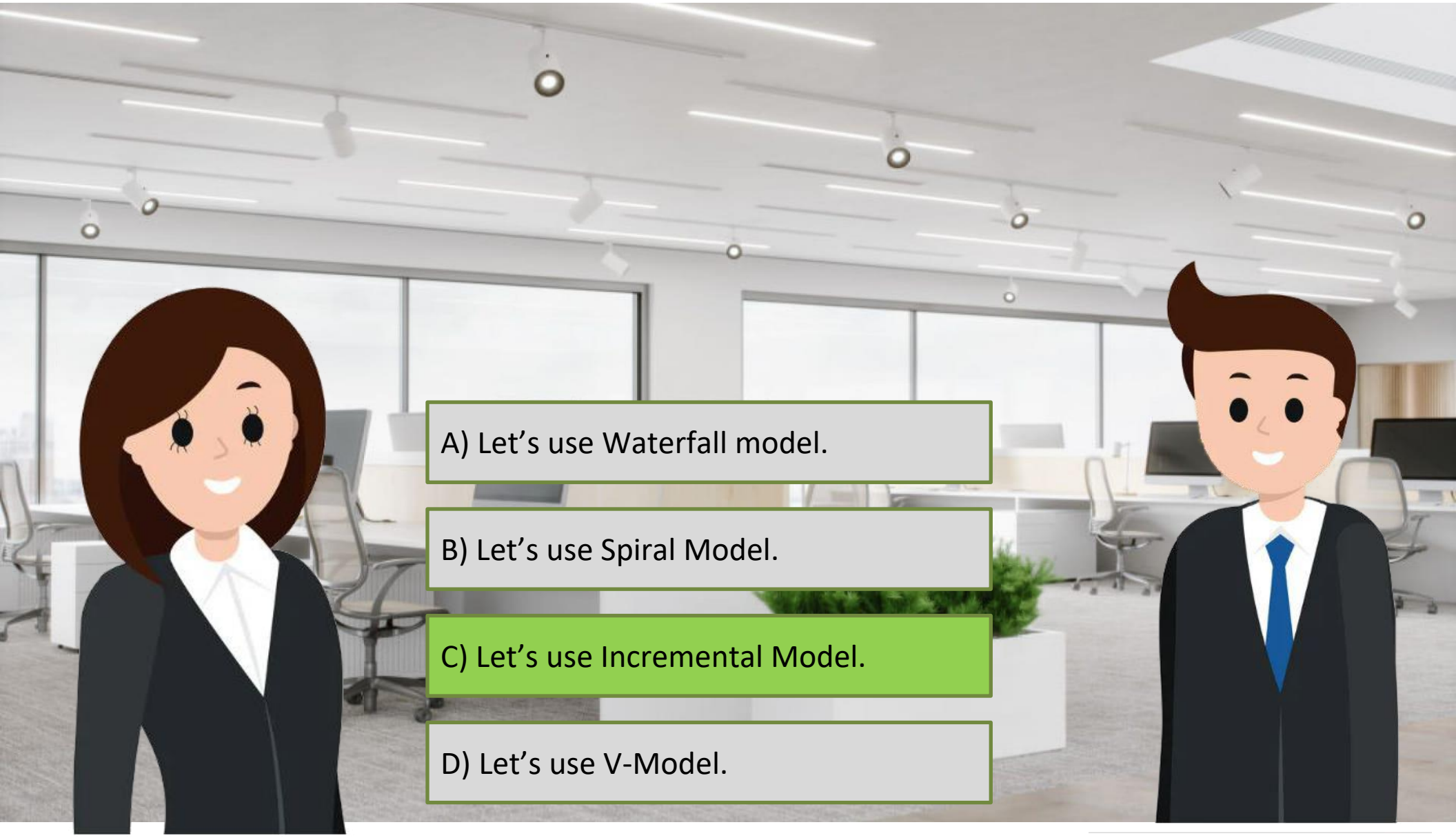

Team Lead: That would not be possible. Did the client say anything else?


Employee: Yes, they said to give them the XYZ feature first and fast.

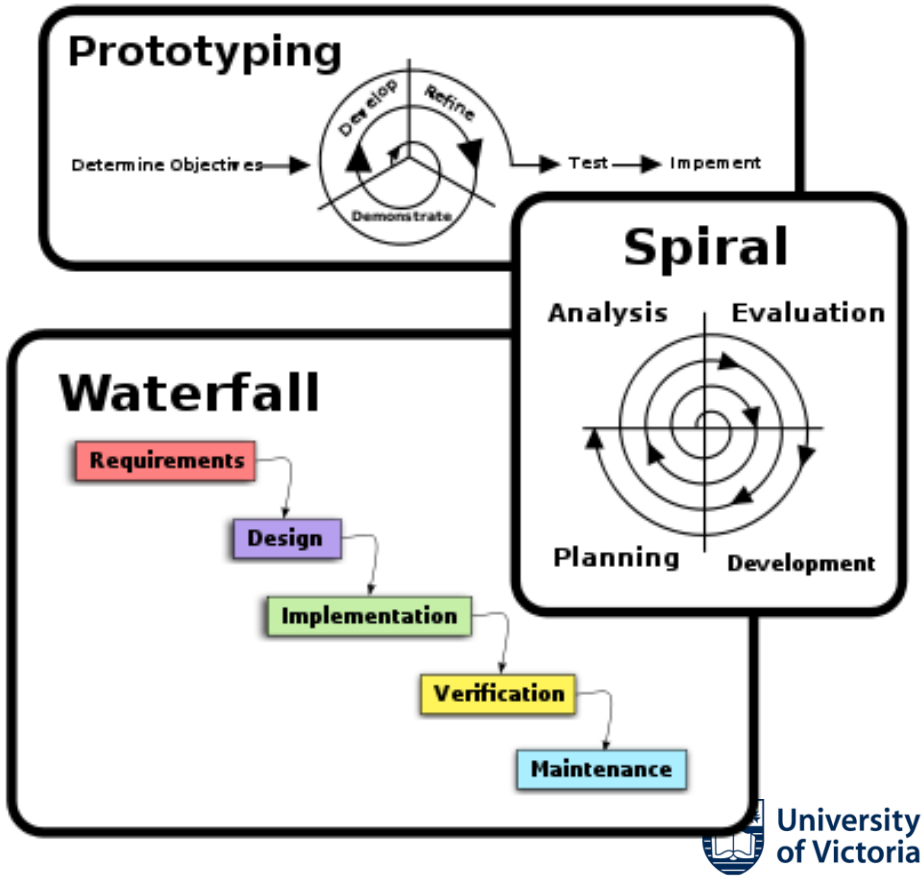

Team Lead: Let’s ask the Project Manager, what to do?


Team Lead: Hi Project Manager, The customer wants the software fast and said that they want this XYZ feature first? What should we do?


A) Let’s use Waterfall model.
B) Let’s use Spiral Model.
C) Let’s use Incremental Model.
D) Let’s use V-Model.


A) Let’s use Waterfall model.
B) Let’s use Spiral Model.
C) Let’s use Incremental Model.
D) Let’s use V-Model.

Case Study 2



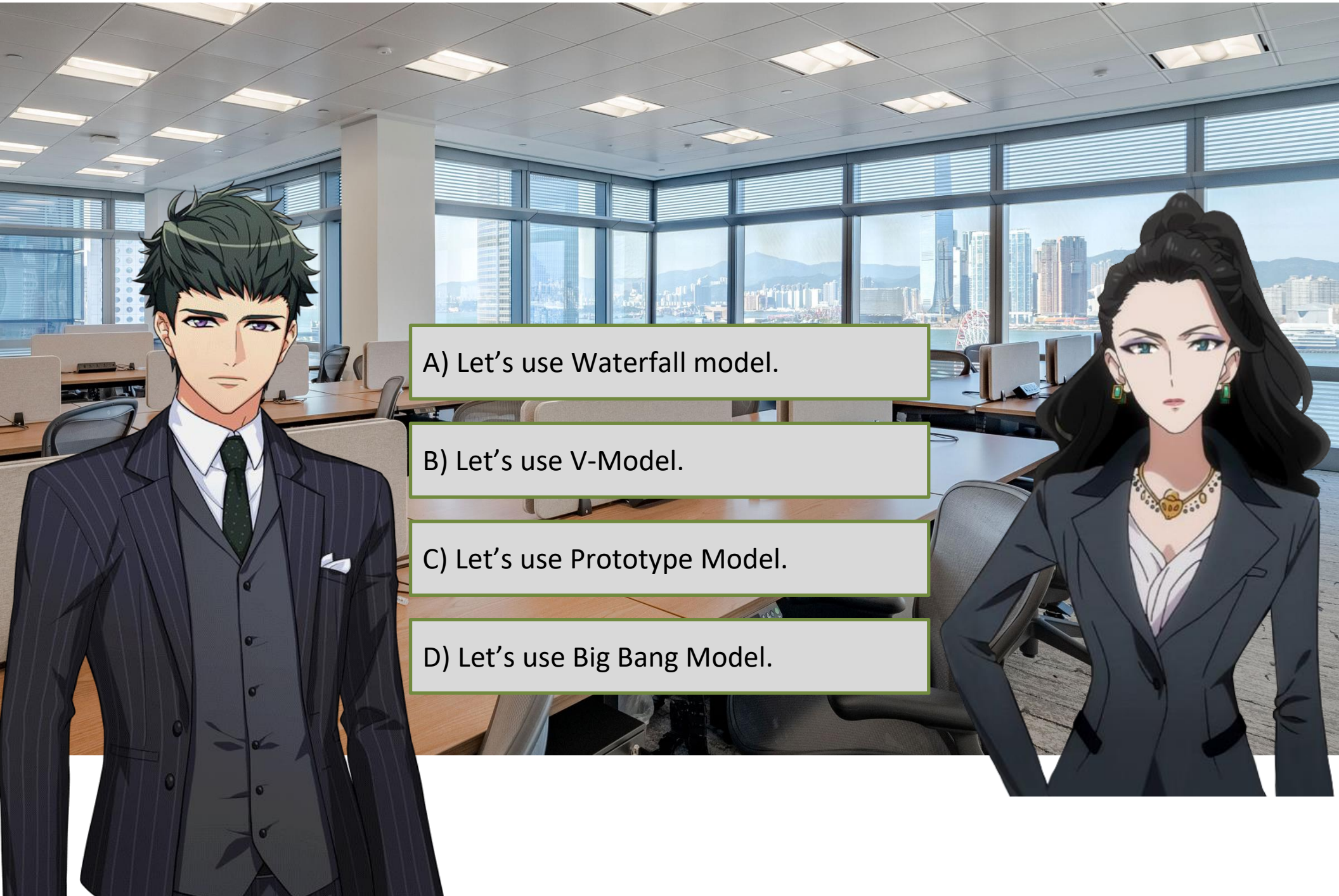
University of Victoria

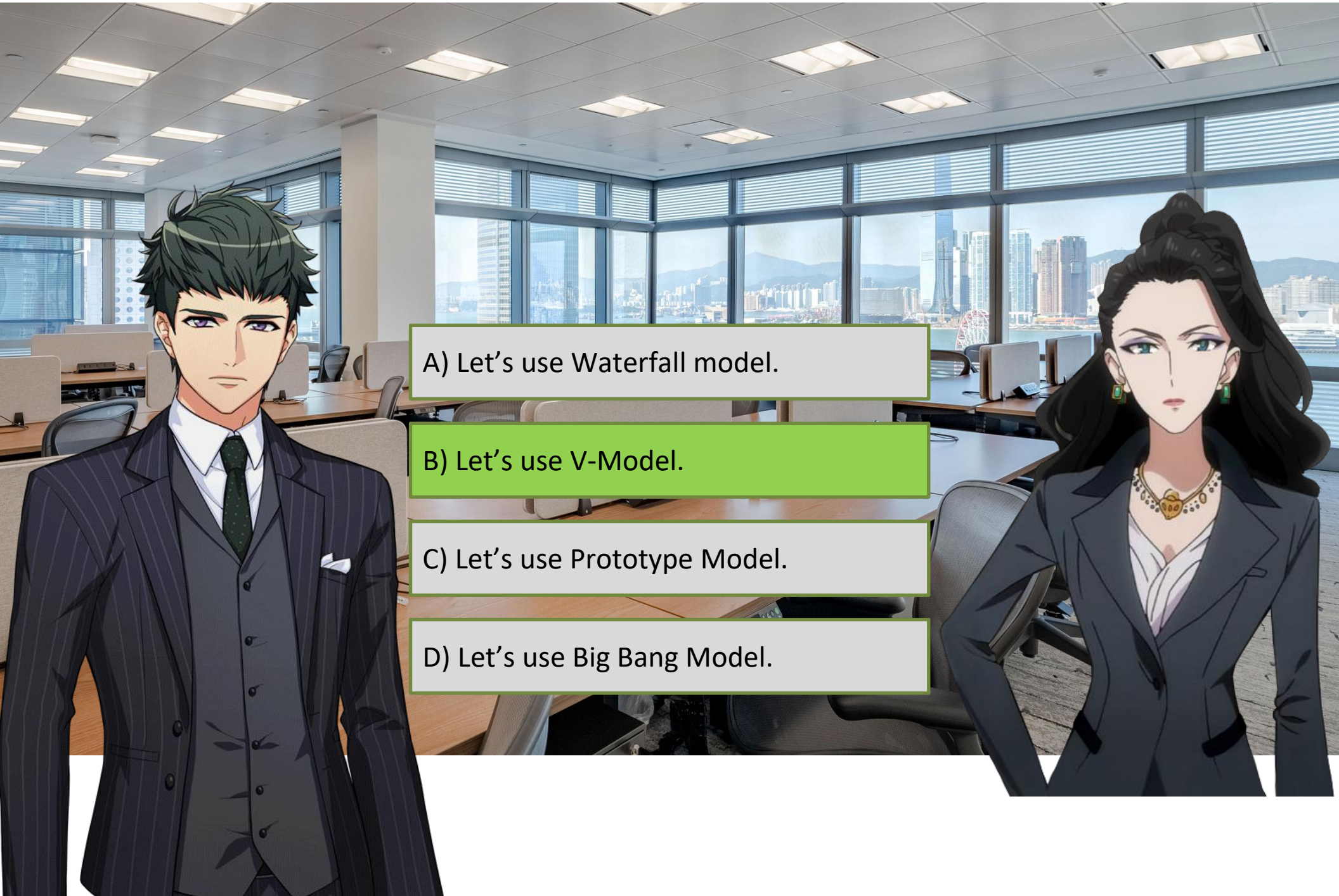

Employee: Team lead, our clients are asking to create the software for new MRI machines?


Team Lead: That’s a serious project. We have full detail on the software’s requirement and our team is full of experts. Let’s ask the project manager which process model to use.


Employee: Sure, Let’s go!!

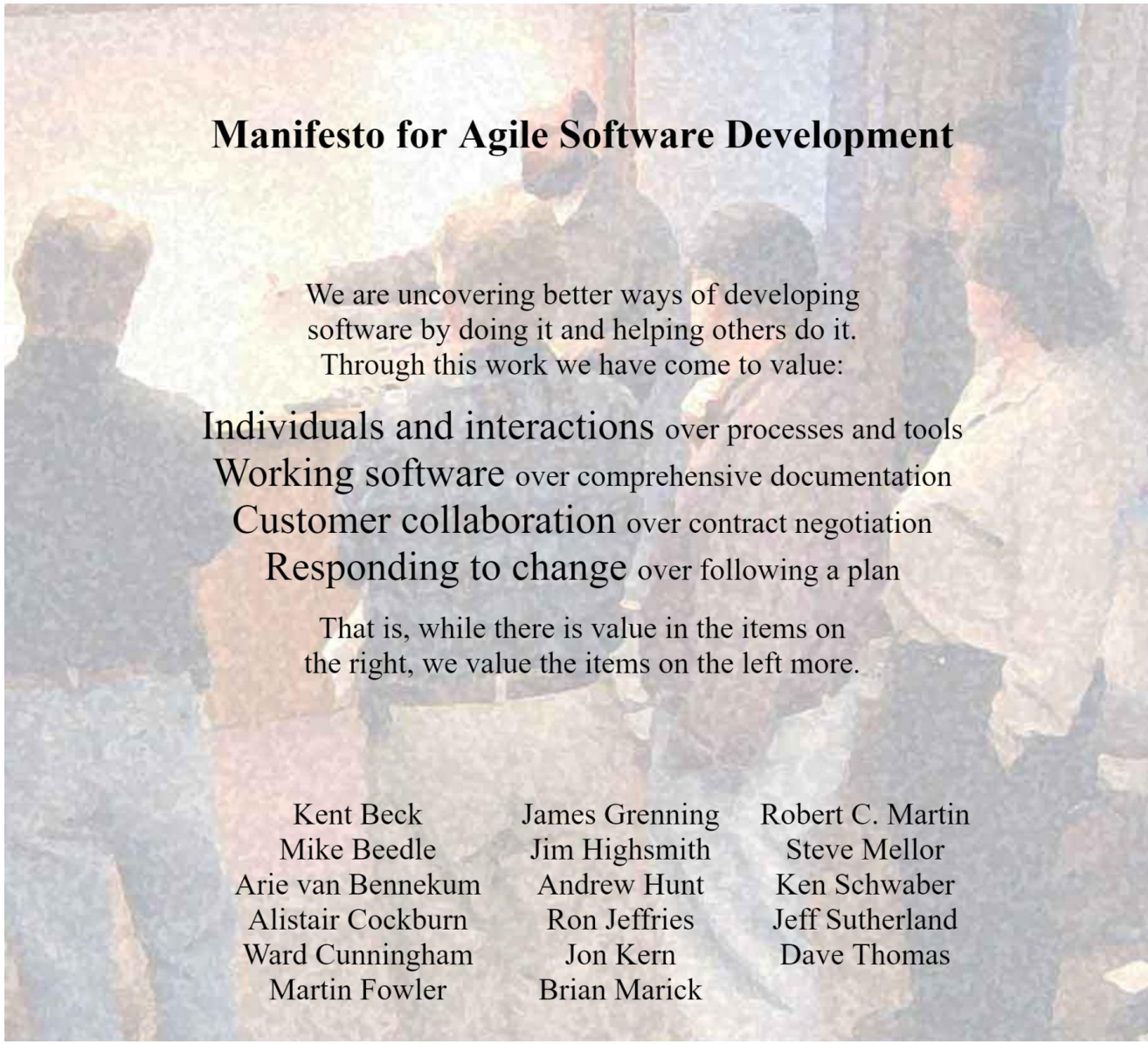

Team Lead: Hi Project Manager, The clients want us to create the software for new MRI machines. We have full detail on the software's requirement and our team is full of experts. Which process model to use ?


A) Let’s use Waterfall model.
B) Let’s use V-Model.
C) Let’s use Prototype Model.
D) Let’s use Big Bang Model.


A) Let’s use Waterfall model.
B) Let’s use V-Model.
C) Let’s use Prototype Model.
D) Let’s use Big Bang Model.

Agile Process Models

Extreme Programming (XP)
Scrum
Dynamic systems development method (DSDM)
Kanban



Software quality

Quality, simplistically, means that a product should meet its specification.
This is problematical for software systems

- There is a tension between customer quality requirements (efficiency, reliability, etc.) and developer quality requirements (maintainability, reusability, etc.);
- Some quality requirements are difficult to specify in an unambiguous way;
- Software specifications are usually incomplete and often inconsistent.

The focus may be ‘fitness for purpose’ rather than specification conformance.

University of Victoria

Software fitness for purpose

- Have programming and documentation standards been followed in the development process?
- Has the software been properly tested?
- Is the software sufficiently dependable to be put into use?
- Is the performance of the software acceptable for normal use?
- Is the software usable?
- Is the software well-structured and understandable?

Software quality attributes

Safety	Understandability	Portability
Security	Testability	Usability
Reliability	Adaptability	Reusability
Resilience	Modularity	Efficiency
Robustness	Complexity	Learnability

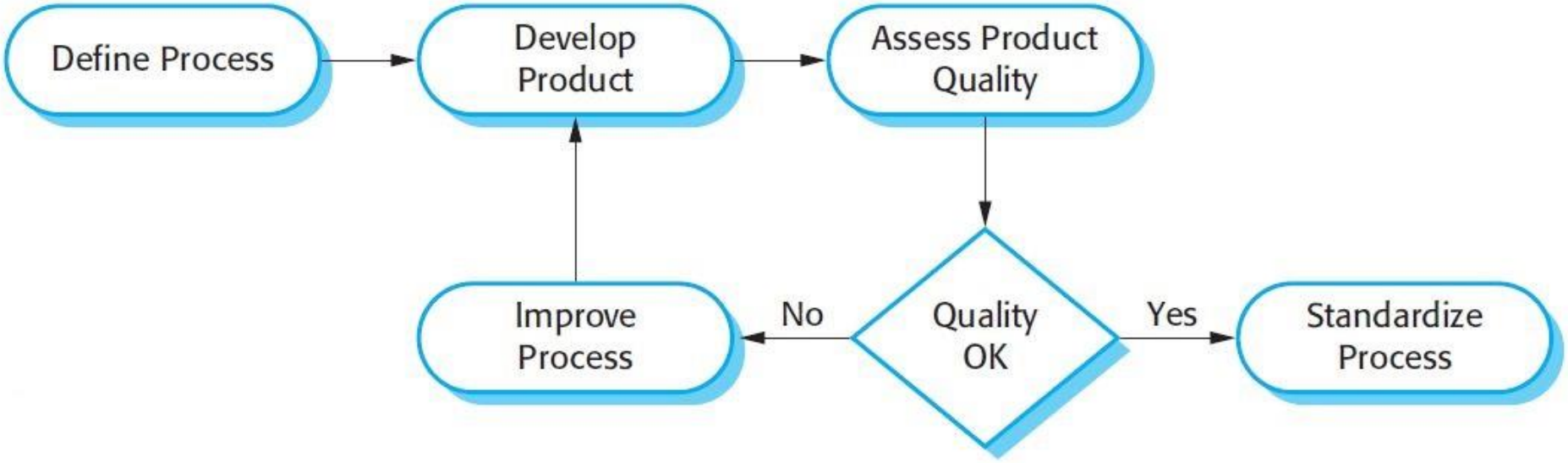
Quality conflicts

- It is not possible for any system to be optimized for all of these attributes.
- The quality plan should therefore define the most important quality attributes for the software that is being developed.
- The plan should also include a definition of the quality assessment process, an agreed way of assessing whether some quality, such as maintainability or robustness, is present in the product.

Process and product quality

- The quality of the production process influences the quality of a developed product.
- This is important in software development as some product quality attributes are hard to assess.
- However, software processes and product quality have a very complex and poorly understood relationship.
  - The application of individual skills and experience is particularly important in software development.

Process-based quality



University of Victoria

Summary

- 4+1 View Model (Scenarios as a glue to relate views)
- Use Case Modelling for Scenarios
- Process Models
- Software Quality

University of Victoria