SOFTWARE ENGINEERING II
SCS 2103

Roshan Rajapakse
University of Colombo School of Computing
rnr@ucsc.cmb.ac.lk

LEARNING RESOURCES

- System Analysis and DESIGN METHODS By Jeffrey L Whitten & Lonnie D Bentley ISBN 0-07-063417-3 (7th Edition)
- SAMs Teach Yourself UML in 24 Hours, Joseph Schmuller, 3rd Edition, ISBN 81-297-0609-1, Pearson Edu.,2004
- Software Engineering,   Ian Somerville, 9th edition , ISBN 978-81-317-6216-5 Pearson , 2011

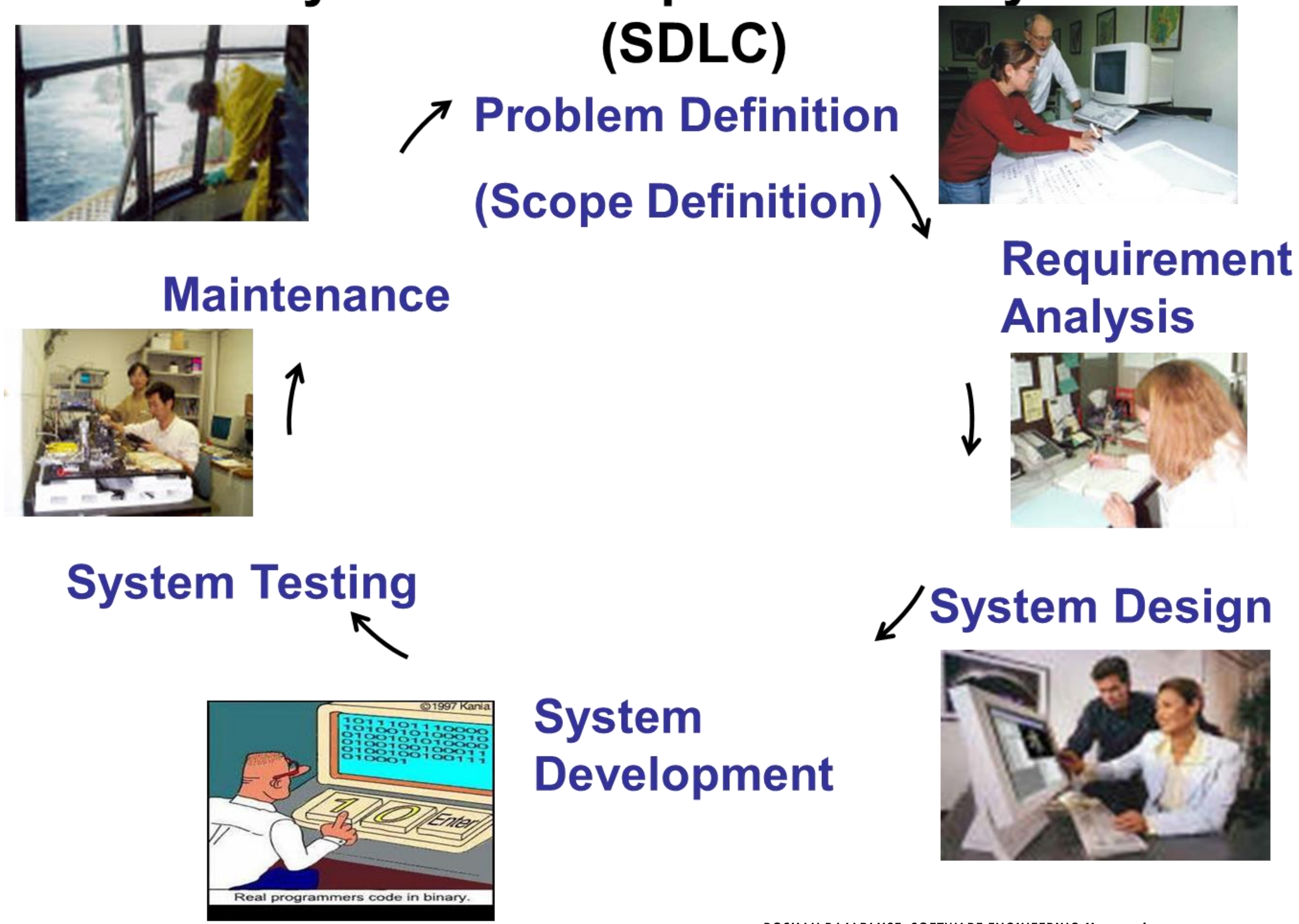
LEARNING OUTCOMES

- Describe Object Oriented Analysis and Design concepts and apply them to solve problems
- Prepare Object Oriented Analysis and Design documents for a given problem
- Introduce and apply some advanced software engineering techniques, beyond those covered in Software Engineering I

- **Software** : Eclipse Papyrus or ArgoUML or Microsoft Visio or Visual Paradiam or any tool that supports UML 1.4 and higher

RECAP

System Development Life Cycle (SDLC)



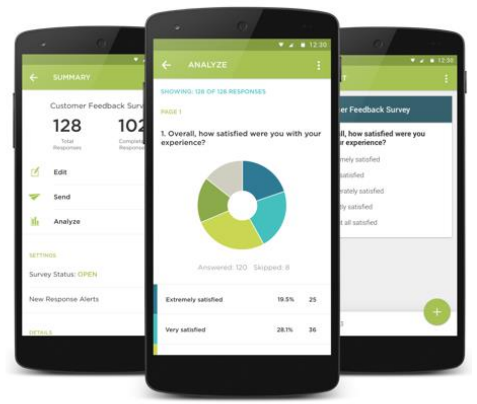
Software development process

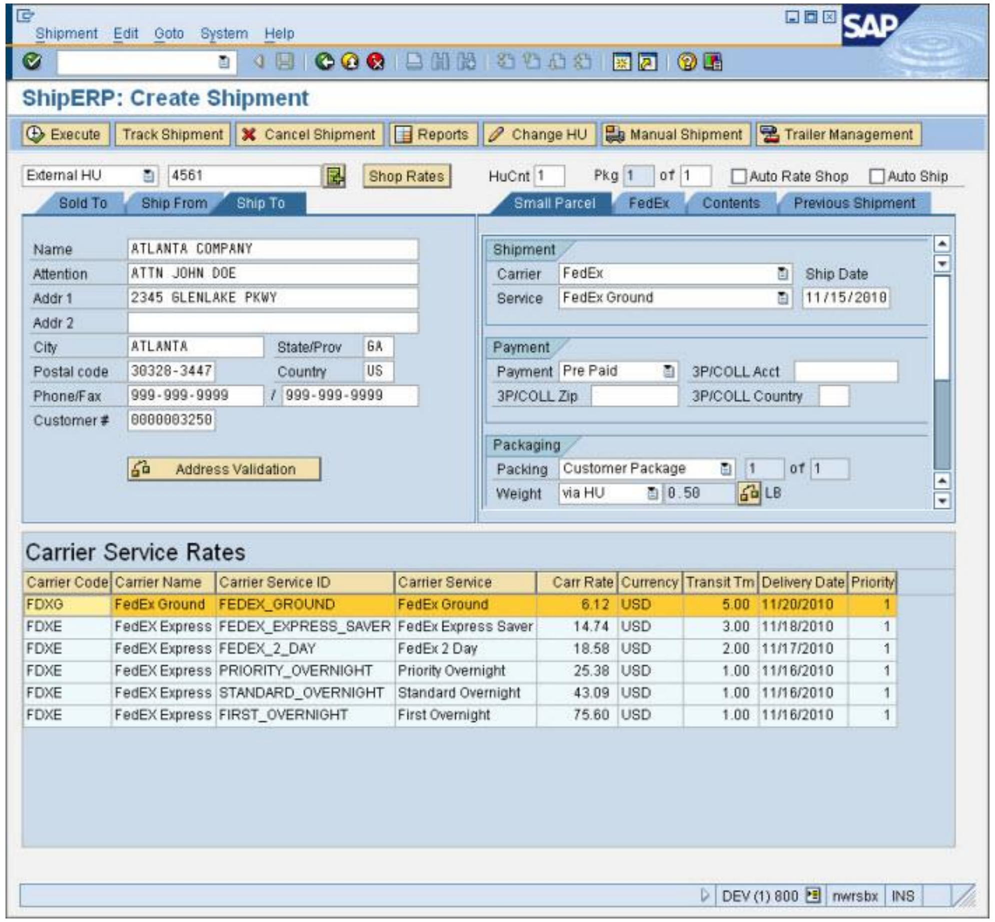
Core activities / The software process	Specification, Design and implementation, Validation, Evolution
Paradigms and models	Software engineering Waterfall Prototyping Incremental V-Model Dual Vee Model Spiral IID Agile Lean DevOps
Methodologies and frameworks	Cleanroom TSP PSP RAD DSDM MSF Scrum Kanban UP XP TDD ATDD BDD FDD DDD MDD
Supporting disciplines	Configuration management Infrastructure as Code Documentation Software Quality assurance (SQA) Project management User experience
Tools	Compiler Debugger Profiler GUI designer Modeling IDE Build automation Release automation Testing

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

SOFTWARE PROCESS?

- Set of related activities that lead to the production of a software product
- Is there; One Process?
- One Road Map?
- One predefine template that will take through all the necessary steps from start to finish to make a piece of software?





ShipERP: Create Shipment

External HU	4561
HuCnt	1
Pkg	1 of 1
Auto Rate Shop	☐
Auto Ship	☐

Name	ATLANTA COMPANY		
Attention	ATTN JOHN DOE		
Addr 1	2345 GLENLAKE PKWY		
Addr 2			
City	ATLANTA	State/Prov	GA
Postal code	30328-3447	Country	US
Phone/Fax	999-999-9999	/	999-999-9999
Customer #	0000003250		

Shipment

Carrier	FedEx	Ship Date	
Service	FedEx Ground		11/15/2010

Payment

Payment	Pre Paid	3P/COLL Acct	
3P/COLL Zip		3P/COLL Country	

Packaging

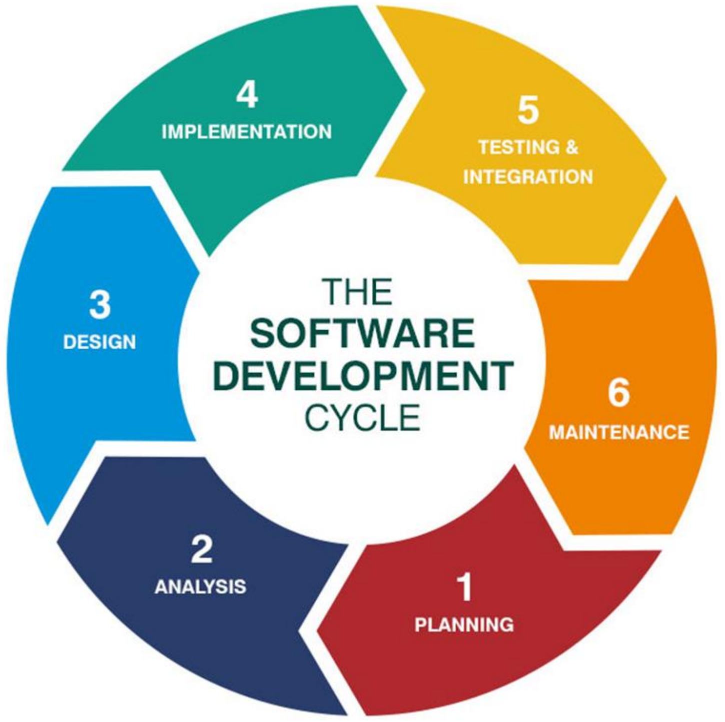
Packing	Customer Package	1	of 1
Weight	via HU	0.50	LB

Carrier Service Rates

Carrier Code	Carrier Name	Carrier Service ID	Carrier Service	Carr Rate	Currency	Transit Tm	Delivery Date	Priority
FDXG	FedEx Ground	FEDEX_GROUND	FedEx Ground	6.12	USD	5.00	11/20/2010	1
FDXE	FedEX Express	FEDEX_EXPRESS_SAVER	FedEx Express Saver	14.74	USD	3.00	11/18/2010	1
FDXE	FedEX Express	FEDEX_2_DAY	FedEx 2 Day	18.58	USD	2.00	11/17/2010	1
FDXE	FedEX Express	PRIORITY_OVERNIGHT	Priority Overnight	25.38	USD	1.00	11/16/2010	1
FDXE	FedEX Express	STANDARD_OVERNIGHT	Standard Overnight	43.09	USD	1.00	11/16/2010	1
FDXE	FedEX Express	FIRST_OVERNIGHT	First Overnight	75.60	USD	1.00	11/16/2010	1

THE SOFTWARE PROCESS

- Many different software processes but all involve:
  - Specification – defining what the system should do;
  - Design and implementation – defining the organization of the system and implementing the system;
  - Validation – checking that it does what the customer wants;
  - Evolution – changing the system in response to changing customer needs.
- Phases will have activities
  - Eg. Establishing a Database
- Activities will have tasks
  - Eg. Writing a piece of source code
- Products/ Roles/ Pre- and post-conditions



THE SOFTWARE PROCESS

1. Plan-driven
   - planned in advance and progress is measured against this plan
2. Agile processes
   - planning is incremental and it is easier to change the process to reflect changing customer requirements

As Boehm and Turner (2003) discuss, each approach is suitable for different types of software. Generally, you need to find a balance between plan-driven and agile processes

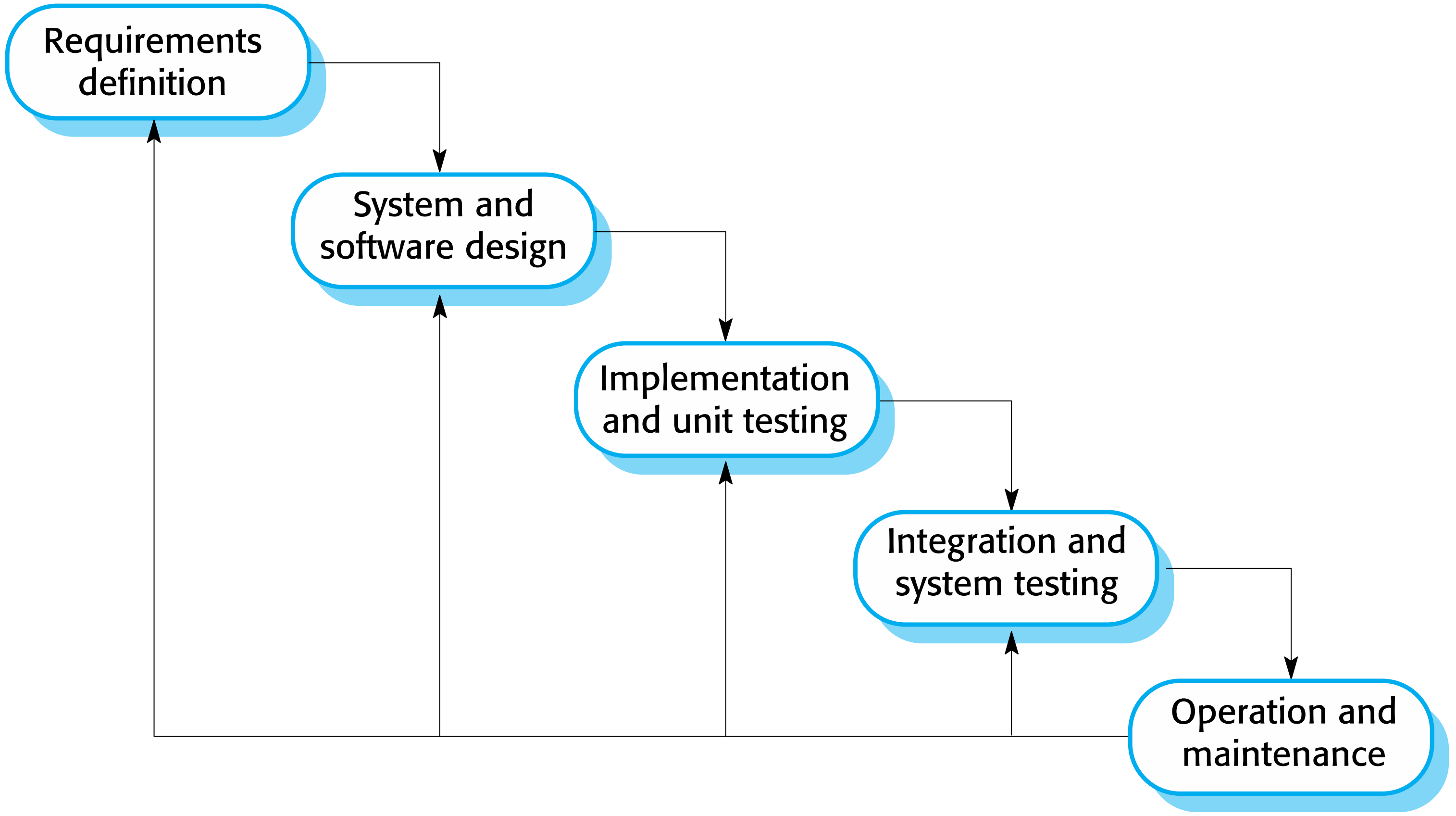
Software development process

Core activities / The software process	Specification, Design and implementation, Validation, Evolution
Paradigms and models	Waterfall Prototyping Incremental V-Model Dual Vee Model Spiral Iterative and incremental development Agile Lean DevOps
Methodologies and frameworks	Cleanroom TSP PSP RAD DSDM MSF Scrum Kanban UP XP TDD ATDD BDD FDD DDD MDD
Supporting disciplines	Configuration management Infrastructure as Code Documentation Software Quality assurance (SQA) Project management User experience
Tools	Compiler Debugger Profiler GUI designer Modeling IDE Build automation Release automation Testing

SOFTWARE PROCESS MODELS

- A simplified representation of a software process
- **The waterfall model**
  - Plan-driven model. Separate and distinct phases of specification and development.
- **Incremental development**
  - Specification, development and validation are interleaved. May be plan-driven or agile.
- **Reuse-oriented software engineering**
  - The system is assembled from existing configurable components. May be plan-driven or agile.

THE WATERFALL MODEL


Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

Software Engineering, Ian Somerville, 9th edition , ISBN 978-81-317-6216-5 Pearson , 2011

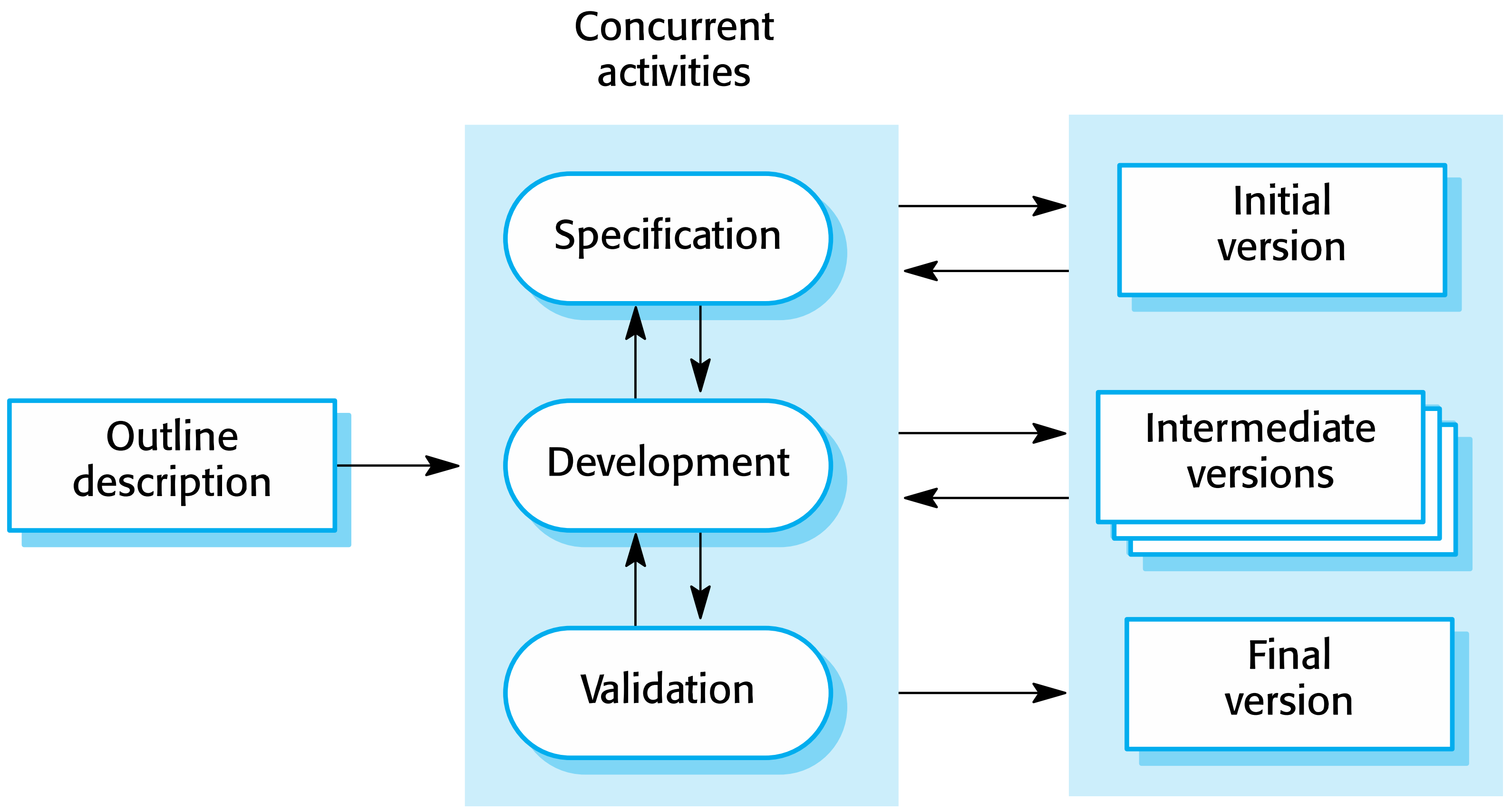
WATERFALL METHOD

- The result of each phase is one or more documents that are approved (‘signed off’)
- In principle, a phase has to be complete before moving onto the next phase
- In practice, these stages overlap and feed information to each other
- Due to costs of producing and approving documents, iterations can be costly and involve significant rework.
- Therefore, after a small number of iterations, it is normal to freeze parts of the development, such as the specification

WATERFALL MODEL PROBLEMS

- <u>Inflexible partitioning of the project into distinct stages </u>->
- Makes it difficult to respond to changing customer requirements.
  - Only appropriate when the requirements are well-understood and changes will be fairly limited during the design process.
- Mostly used for large systems engineering projects where a system is developed at several sites.
  - the plan-driven nature of the waterfall model helps coordinate the work.
- **Should only be used when the requirements are well understood and unlikely to change radically during system development**

INCREMENTAL DEVELOPMENT



Software Engineering, Ian Somerville, 9th edition , ISBN 978-81-317-6216-5 Pearson , 2011

INCREMENTAL DEVELOPMENT

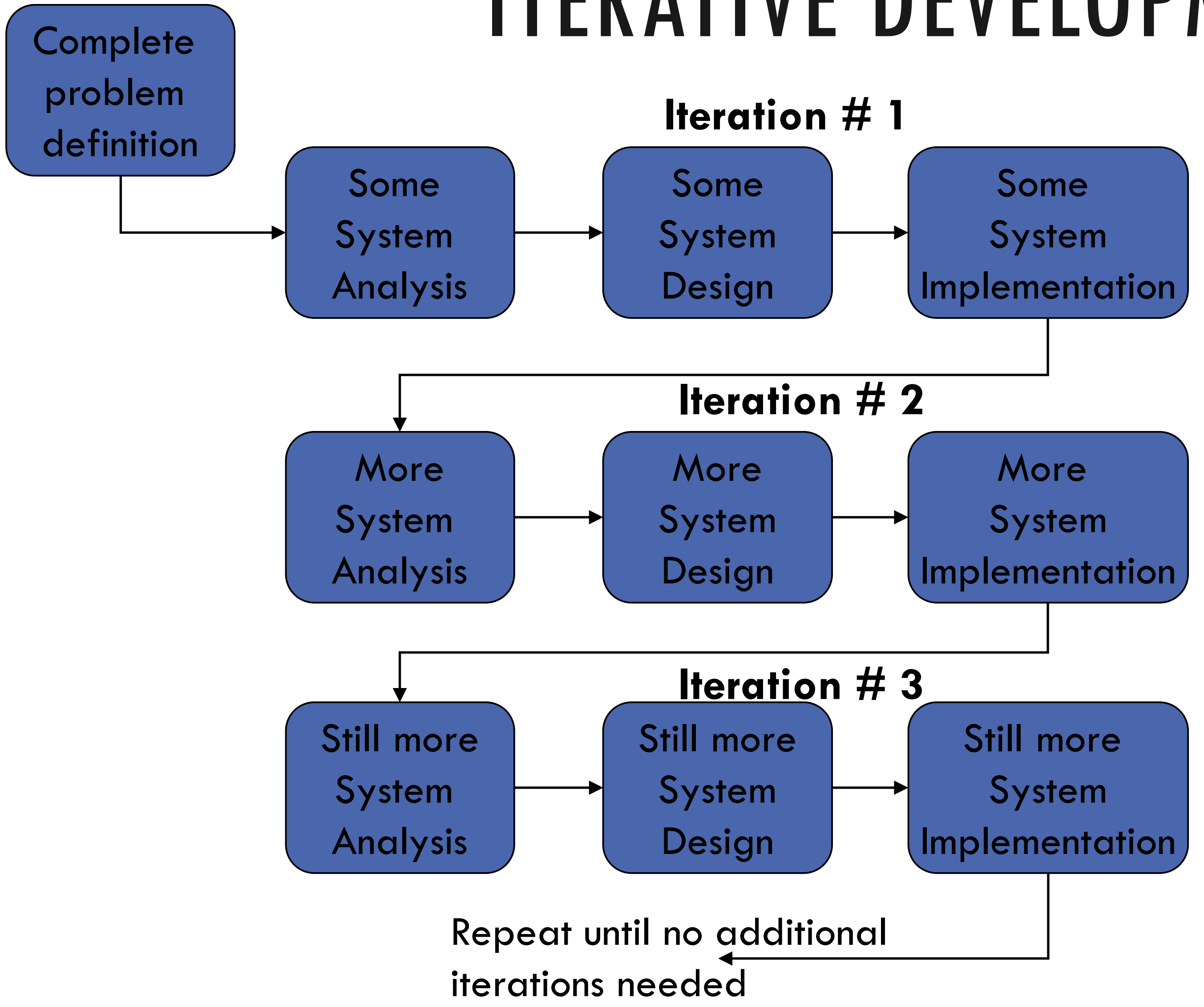
- Rather than delivering the system as a single delivery, the development and delivery is broken down into **increments** with each increment delivering part of the required functionality.
- User requirements are prioritized and the highest priority requirements are included in early increments
- Each component is delivered to the client when it is complete.
- This model of development also helps ease the traumatic effect of introducing a completely new system all at once.
- **The incremental model applies the waterfall model incrementally**

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

INCREMENTAL DEVELOPMENT BENEFITS

- The cost of accommodating changing customer requirements is reduced.
- It is easier to get customer feedback on the development work that has been done.
- More rapid delivery and deployment of useful software to the customer is possible.

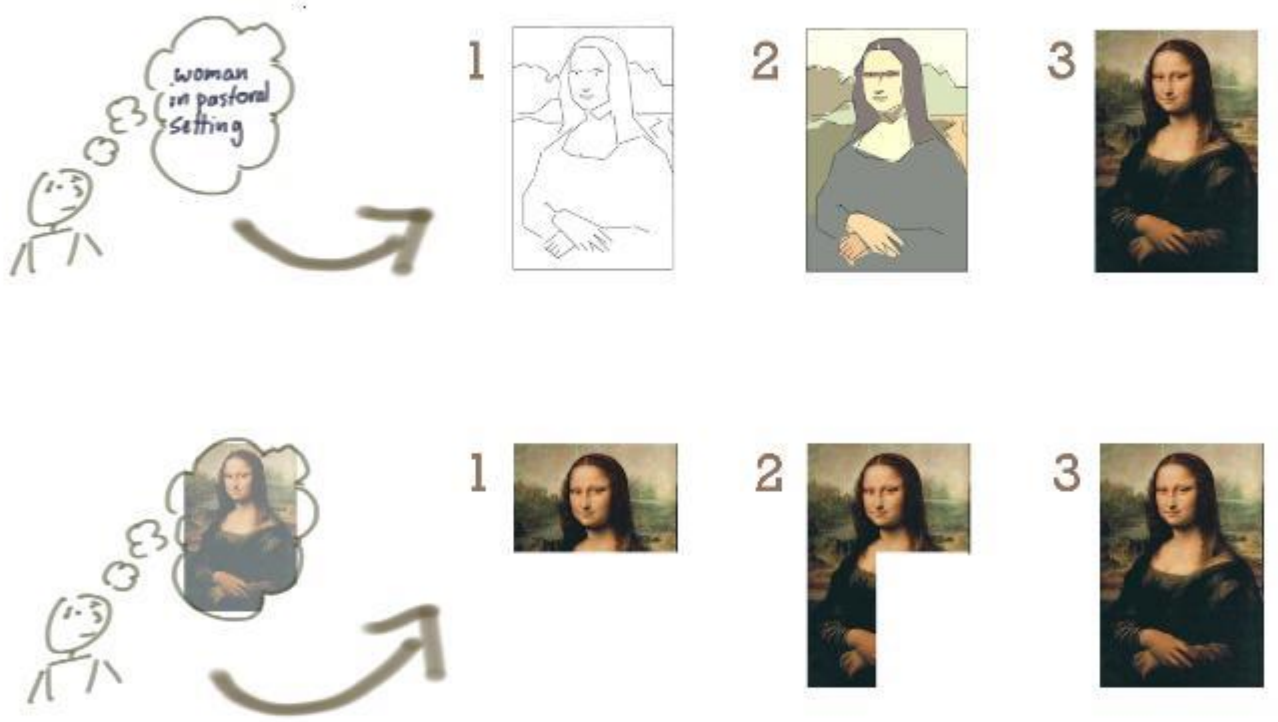
ITERATIVE DEVELOPMENT


Complete problem definition
Iteration # 1
Some System Analysis
Some System Design
Some System Implementation
Iteration # 2
More System Analysis
More System Design
More System Implementation
Iteration # 3
Still more System Analysis
Still more System Design
Still more System Implementation
Repeat until no additional iterations needed

A design methodology based on a cyclic process of prototyping, testing, analyzing, and refining a product or process.

Based on the results of testing the most recent iteration of a design, changes and refinements are made

ITERATIVE VS. INCREMENTAL?



https://watirmelon.blog/2015/02/02/iterative-vs-incremental-software-development/

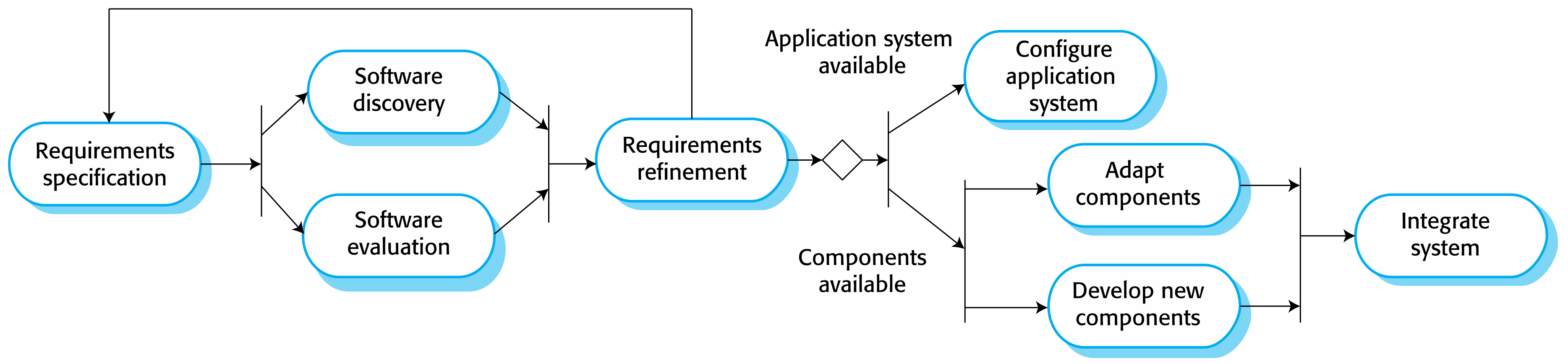
ITERATIVE VS. INCREMENTAL?

- Scrum and agile are both incremental and iterative.
- They are iterative in that they plan for the work of one iteration to be improved upon in subsequent iterations.
- They are incremental because completed work is delivered throughout the project.

INTEGRATION AND CONFIGURATION / REUSE-ORIENTED SOFTWARE ENGINEERING

- Based on software reuse where systems are integrated from existing components or application systems (sometimes called COTS -Commercial-off-the-shelf) systems).
- Reused elements may be configured to adapt their behaviour and functionality to a user's requirements
- Reuse is now the standard approach for building many types of business systems/ scientific systems

REUSE-ORIENTED SOFTWARE ENGINEERING



Software Engineering, Ian Somerville, 9th edition , ISBN 978-81-317-6216-5 Pearson , 2011