COURSE NAME

SOFTWARE ENGINEERING

CSC 3114

(UNDERGRADUATE)

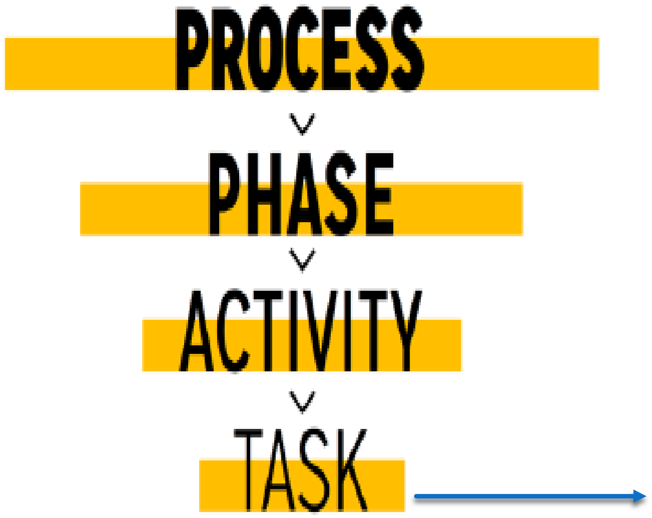
CHAPTER 2

SOFTWARE DEVELOPMENT PROCESS MODEL

M. MAHMUDUL HASAN
ASSISTANT PROFESSOR, CS, AIUB
Web: http://cs.aiub.edu/profile/m.hasan
WordPress: https://wordpress.com/view/mhasansuman.wordpress.com

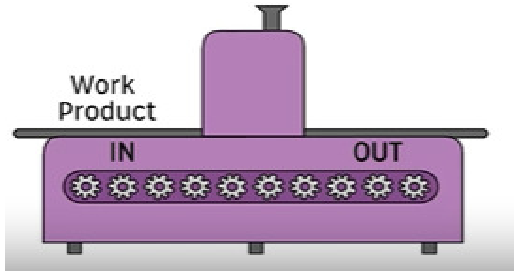
RG Google Scholar LinkedIn

SOFTWARE PROCESS



- **Process** is a way to organize the development of a software product into distinct phases or stages (entire life of a software product or can be a sub-process within a life cycle process.
- **Phase:** processes are organized into phases. For Examples: the lifecycle process is organized as specification, design and implementation, and verification and validation phases.
- **Activity:** each phase is composed of associated activities.
- **Task:** an activity is a group of related tasks (low-level actions)
- **Role:** and tasks relate to roles, work products, and resources.

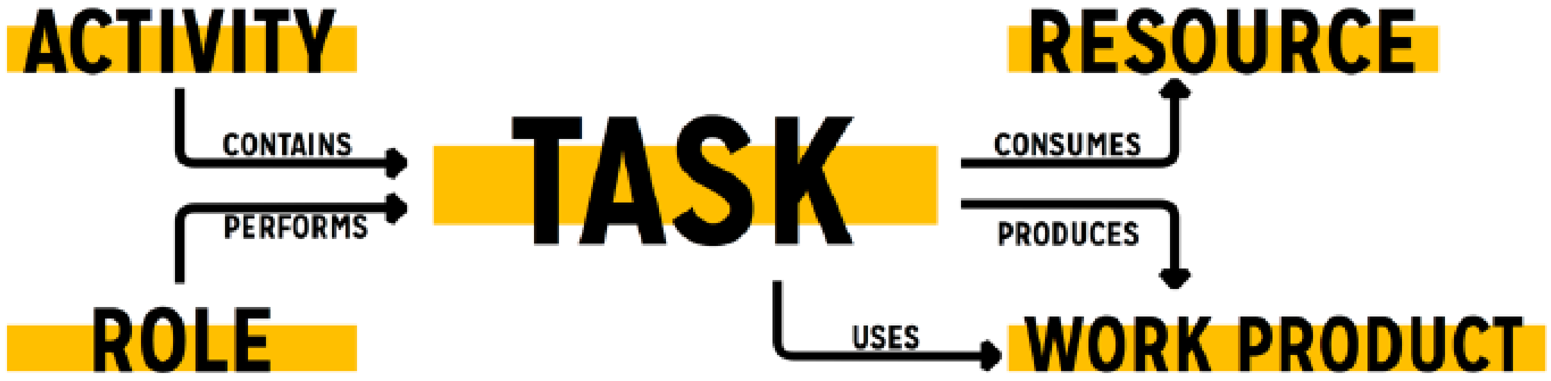
❑ **Practices:** practices are guidelines that can be applied within a process with proven benefits such as increased communication, tracking of project progress, risk management, and reduced waste. Example: Agile methodologies.



- An Activity (Design) takes a work product as input (Specification) and produce another work product as output (UI/UX)

SOFTWARE DEVELOPMENT PROCESS MODELS

- Completing any given task requires resources to advance or fund it, such as time, people, technology, knowledge, and money.
- *A task* ***consumes*** *resources.*
- *resources* ***consume by a*** *task*



SOFTWARE DEVELOPMENT PROCESS MODELS

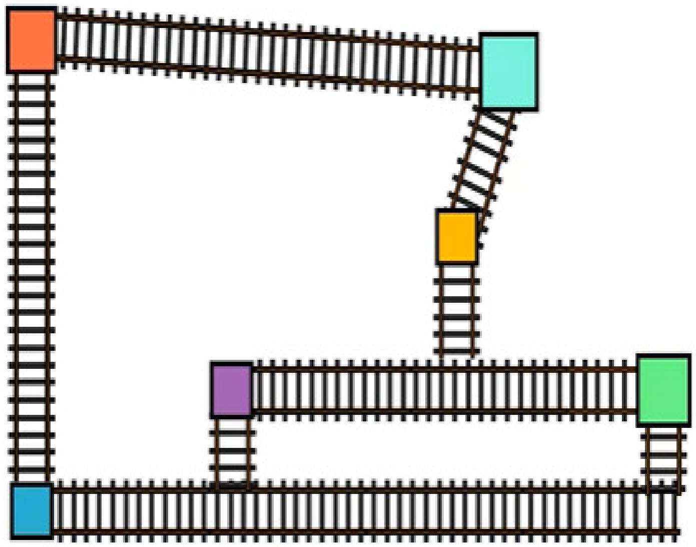
- A variety of process models will be presented during this course
- Every process model will have phases; however, there are major differences between models in terms of what activities constitute their phases and the sequencing of the phases.
- Three main types of process models are:
- **Linear process models** – phases that happen sequentially, one after another
- **Incremental process models –** deliver the software through several increments
- **Iterative process models** – phases that are repeated in cycles
- **Parallel process models** – activities that occur concurrently

SOFTWARE DEVELOPMENT PROCESS MODELS

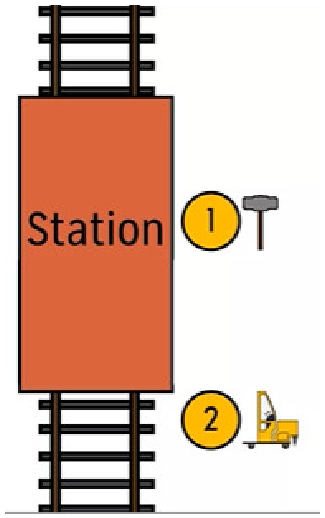
Q. Why Do You need to Learn all the available Process Models for Software development?

Technology A: Automated, Costly

Technology B: Manual, Less Costly

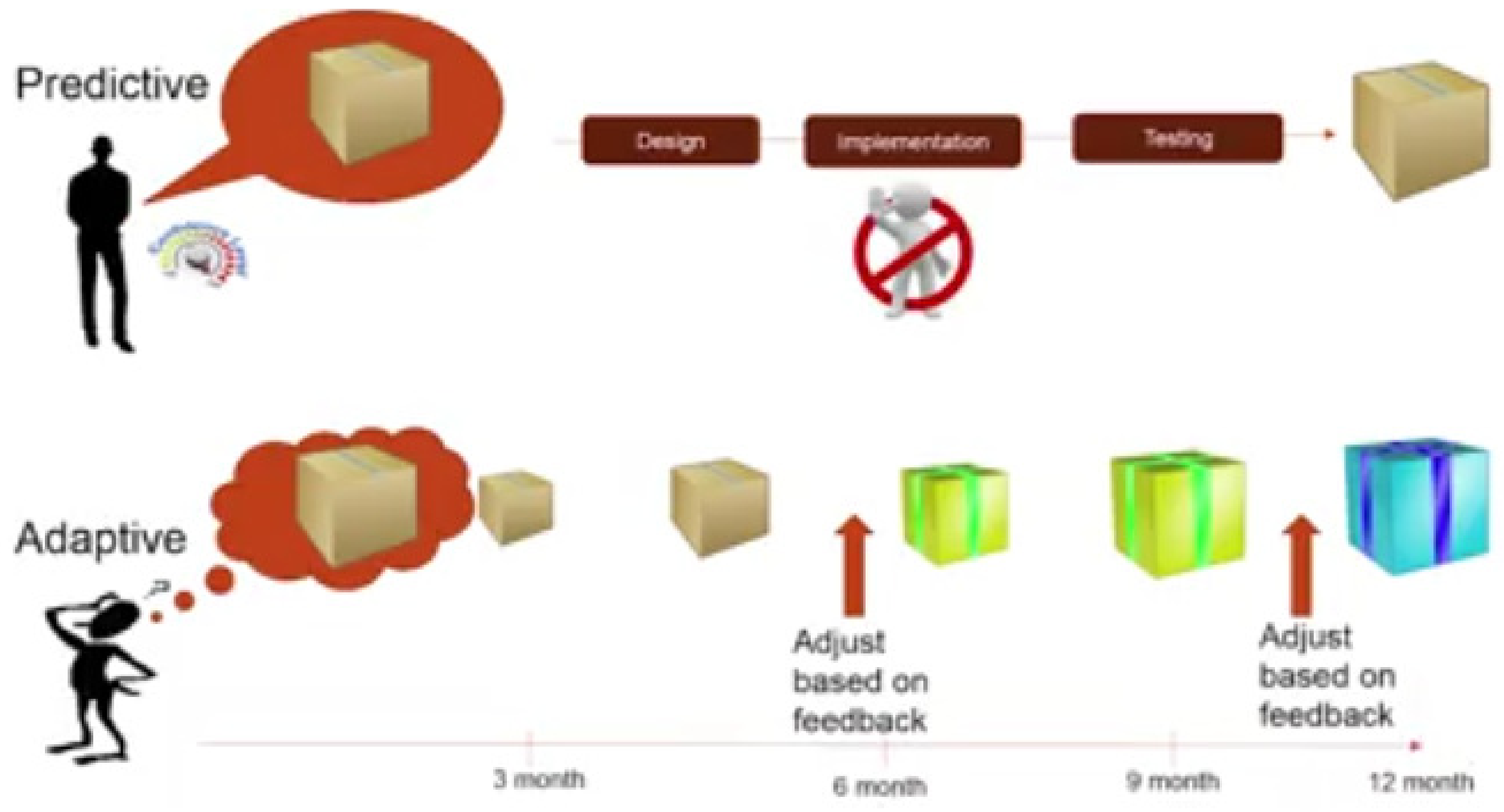


Project 1: Setup a new railway network



Project 2: Repair 100-meter railway track

PROJECT ENVIRONMENT: PREDICTIVE VS ADAPTIVE



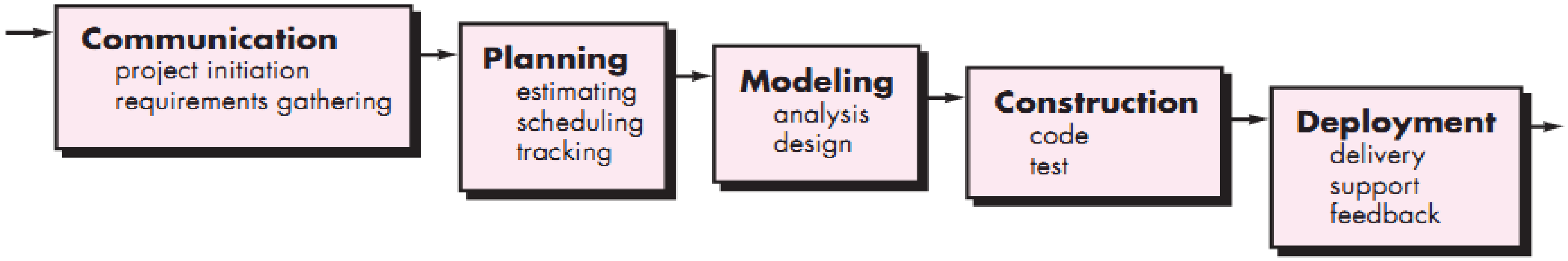
NO Change in the development process

ALLOW Changes at the middle of the development

WATERFALL MODEL : PROCESS

❑ Define the Waterfall model and its stages

- Waterfall model is a linear sequential model where next phase starts only after completion of the previous phase
- It is very simple to understand and easy to use because of its systematic well-defined process description. The product is delivered to the users at ONE shoot (Big bang)



WATERFALL MODEL : IMPACT

❑ Describe its impacts in various software development context

- Environmental impact: this model is very useful and inexpensive for stable environment where the requirements are fixed (i.e., predictive environment) and expensive for dynamic environment where the requirements are frequently changing (i.e., adaptive environment)
- Scope of change: the use of waterfall model makes the software system very rigid for further modification
- Risk lifetime: the risk remains high in long time the project and addresses the maximum risk at the later stages of software development

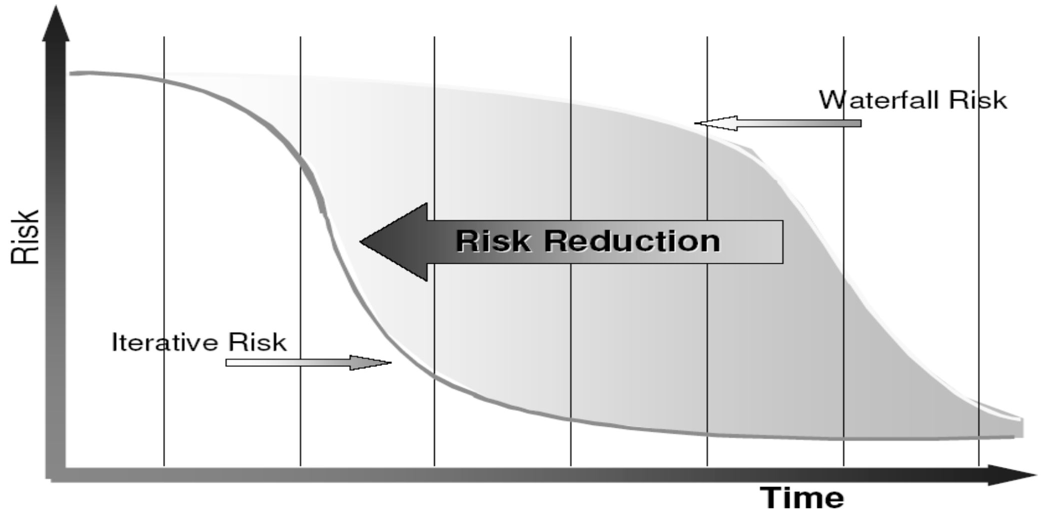


Figure: Risk profile of Software Development Process Model

WATERFALL MODEL : APPLICABILITY

❑ Describe the applicability of waterfall model

- Stable Environment (Predictive): this model is applicable where the software development environment is relatively stable and very little or no scope for any further changes during the development and after releasing the product
- Well-understood Requirements: this model is only appropriate when the requirements are well-understood and no further changes are required in the software descriptions
- Example: in the safety critical product development where the requirements have to well defined at the beginning of the project development

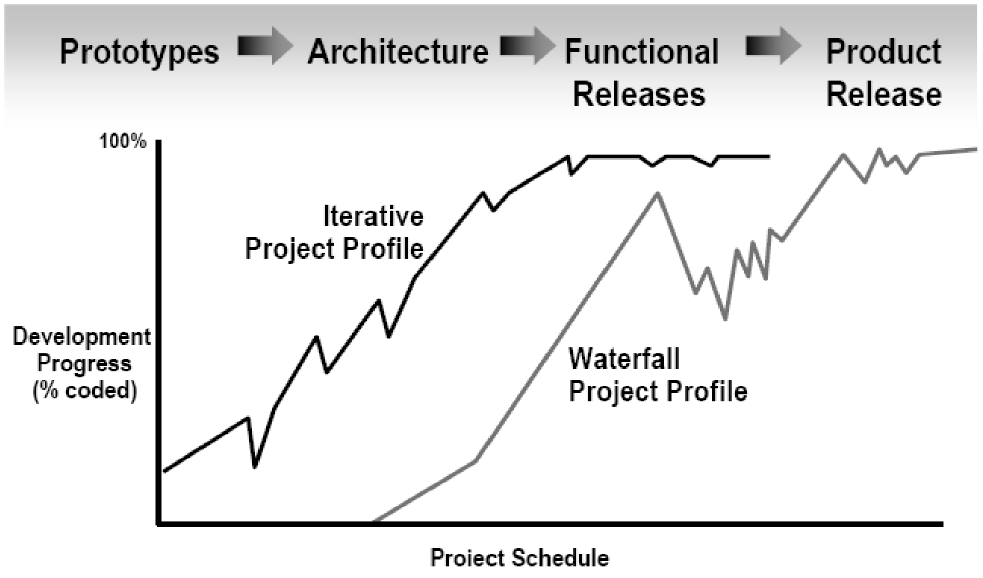
WATERFALL MODEL : EXAMPLE

- Provide some real-life software development examples where waterfall model can be used a process model

- Example: Autopilot system, Chemical plant system, Embedded system where requirements are well understood and no need for further modification of the system specification

WATERFALL MODEL : LIMITATION

- ❑ Describe problems of waterfall model
- No Backtracking: Inflexible partitioning of the project into distinct stages where next phase starts only after completion of the previous phase, this makes it difficult to respond to changing customer requirements.
- High Risk: risk remain high longer period in the project and addresses maximum risk at the later stages of software development
- Late Product Visibility: the customer must have patience. A working version of the program(s) will not be available until late in the project time-span.

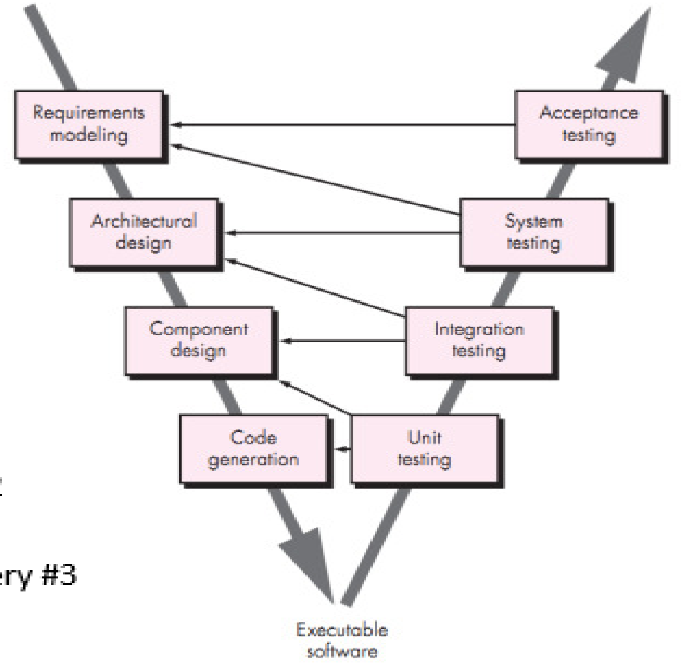


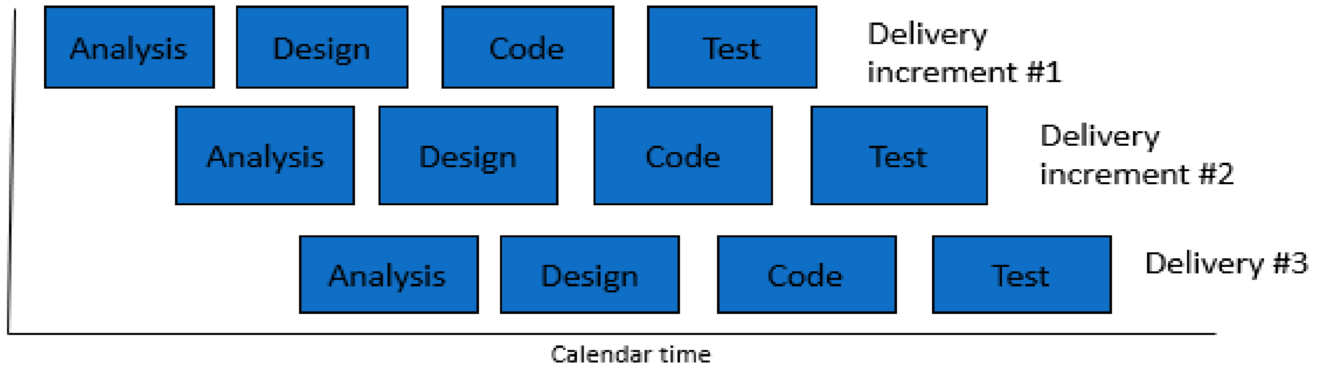
- ❑ Iterative Development: products are visible at an early stage of development. Low probability of rework in case of defects in the final deliverable product

EXTENSION OF WATERFALL MODEL

❑ How some other models can use waterfall model in its process to adapt changes

- V-Model: it is an extension of the waterfall model and is based on association of a testing phase for each corresponding development stage of waterfall model.



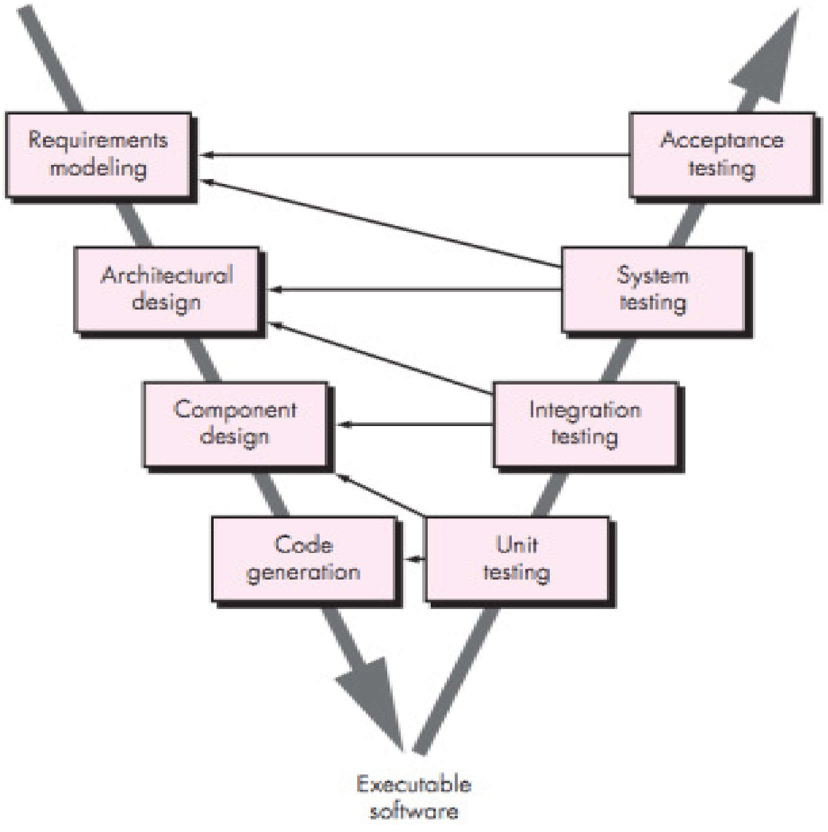


- Incremental Model: the development and delivery are broken down into increments (timeboxes) where each increment may follow the sequential stages of waterfall model for complete one increment without implementing any change in that increments

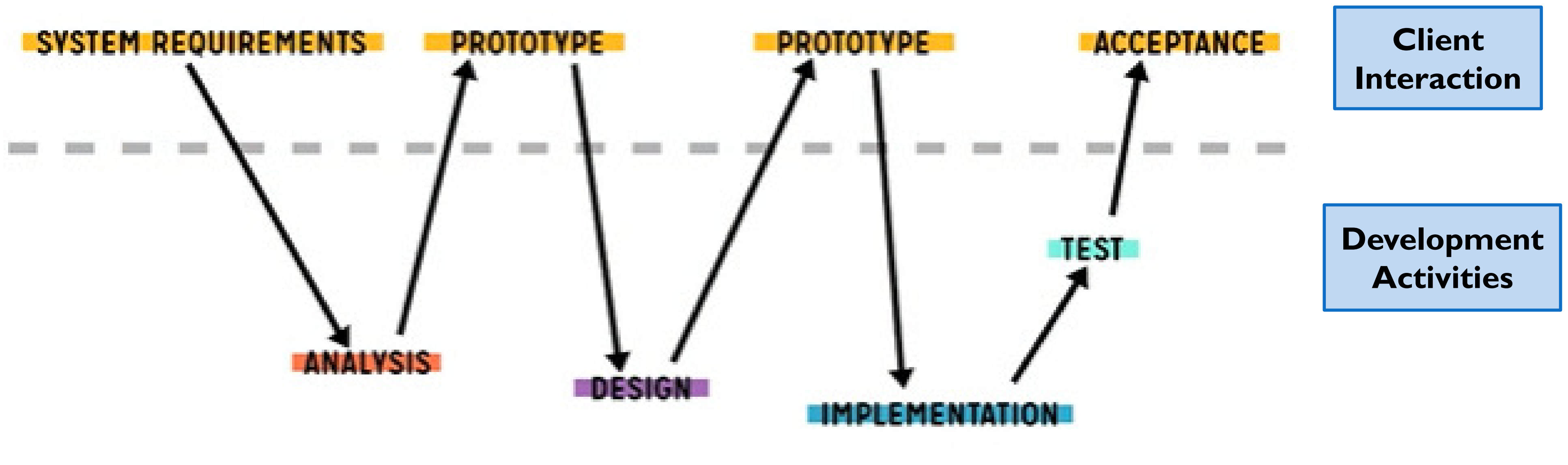
V - MODEL

❑ **How does V-Model Extends Waterfall Model**

- The V-model is a SDLC model where execution of processes happens in a sequential manner in V-shape. It is also known as Verification and Validation model.
- V-Model is an extension of the waterfall model and is based on association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle there is a directly associated testing phase.
- This is a highly disciplined model and next phase starts only after completion of the previous phase.



SAW-TOOTH MODEL

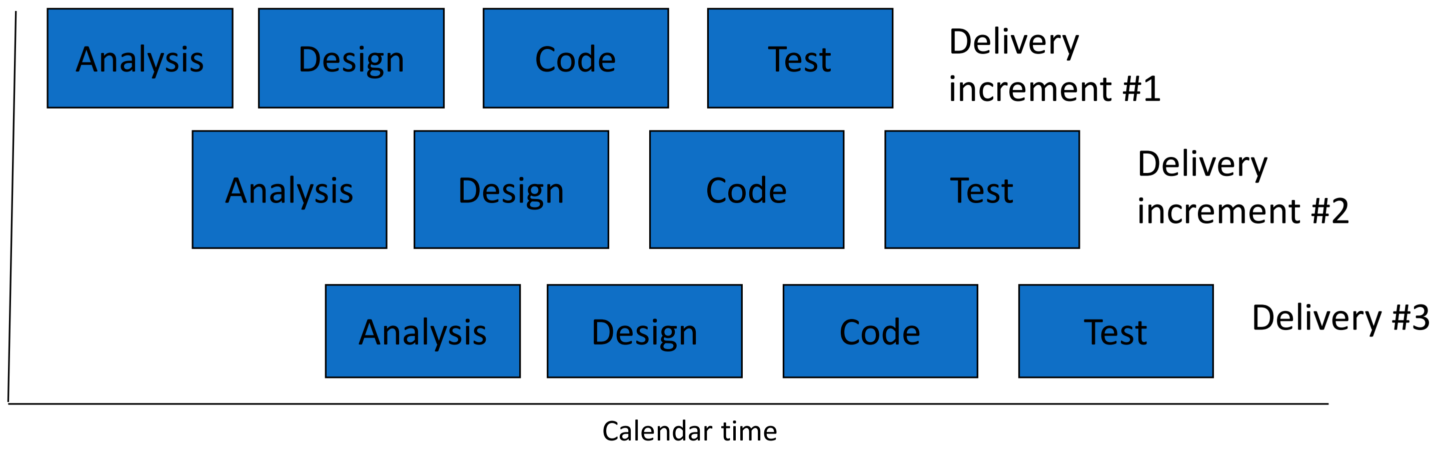


- The Sawtooth process model distinguish the process phases into client side (upper) and development side (bottom) activities
- Waterfall, V-Model, and Sawtooth models are linear sequential model and has same kind of drawbacks in software development.

INCREMENTAL DEVELOPMENT

- Rather than deliver the system as a single delivery, the development and delivery is broken down into increments with each increment delivering part of the required functionality (SPIRAL). The requirements are relatively certain but there are many complexities that leads to frequent changes.
- User requirements are prioritised, and the highest priority requirements are included in early increments
- Once the development of an increment is started, the requirements are frozen though requirements for later increments can continue to evolve

Incremental and Parallel Development



INCREMENTAL DEVELOPMENT

❑ Advantages of Incremental Development

- Customer value can be delivered with each increment so system functionality is available earlier
- Deliver the core product first
- The highest priority system services tend to receive the most testing
- Early increments act as a prototype to help elicit requirements for later increments
- Lower risk of overall project failure

❑ Disadvantages of Incremental Development

- Often need to rework of the components (e.g., time, budget)

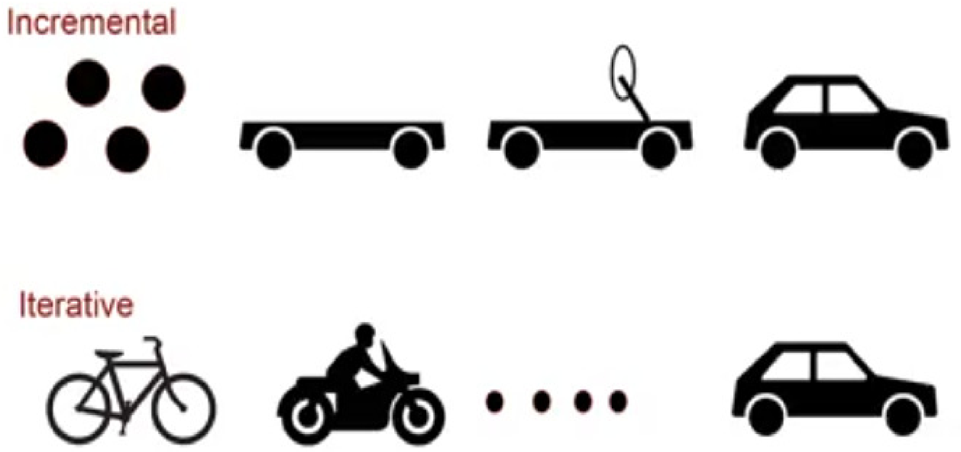
INCREMENTAL VS ITERATIVE

❑ Incremental Development

- **Incremental** development slices work into small bite-sized pieces. These are called increments. Each increment builds on top of what has gone before. Fully functional modules grow bit-by-bit over time. Each evolution adds to preceding functionality.

- The mini **Waterfall Model** is a traditional incremental development approach. Other models can also be used in each increments such as V model, Saw Tooth model

- The **Agile Modeling** is a typical iterative approach.

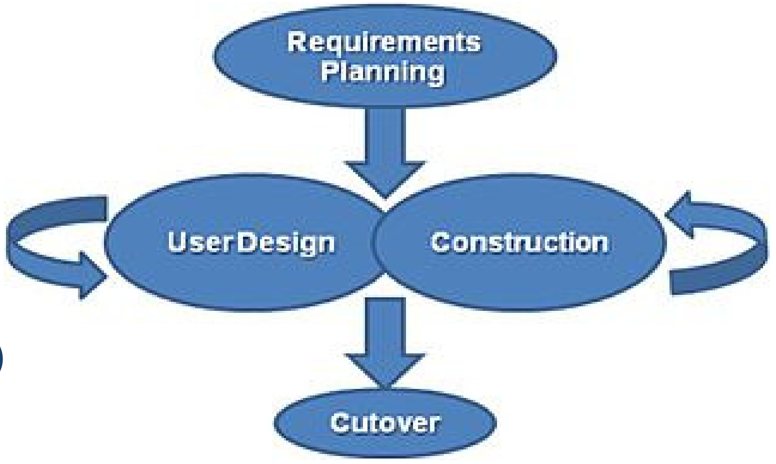


❑ Iterative Development

- **Iterative** development is the process of repeating and refining a cycle/way of working. This is called an iteration.

ITERATIVE: RAPID APPLICATION DEVELOPMENT (RAD)

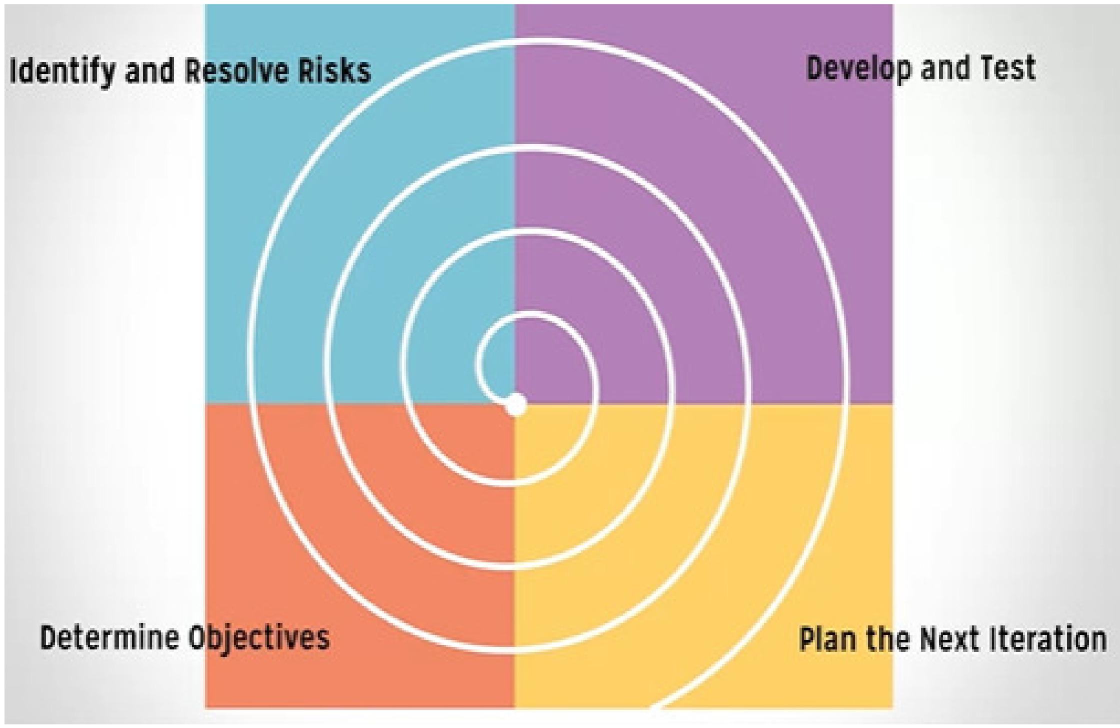
- It is a type of iterative model. The developments are time boxed, delivered and then assembled into a working prototype
- In RAD model the components or functions are developed in parallel as if they were mini projects (frozen requirements in each increments)
- This can quickly give the customer something to see and use and to provide feedback
- Delivers a fully functional system in 90 days, give or take 30 days
- Phases of RAD are:
  - Requirements Planning (user requirements)
  - User Design (user interact with the system analysts)
  - Construction (program and application development)
  - Cutover (testing, changeover to new system, user training)



SPIRAL: AN ITERATIVE MODEL

Each quadrant is a phase of an iteration

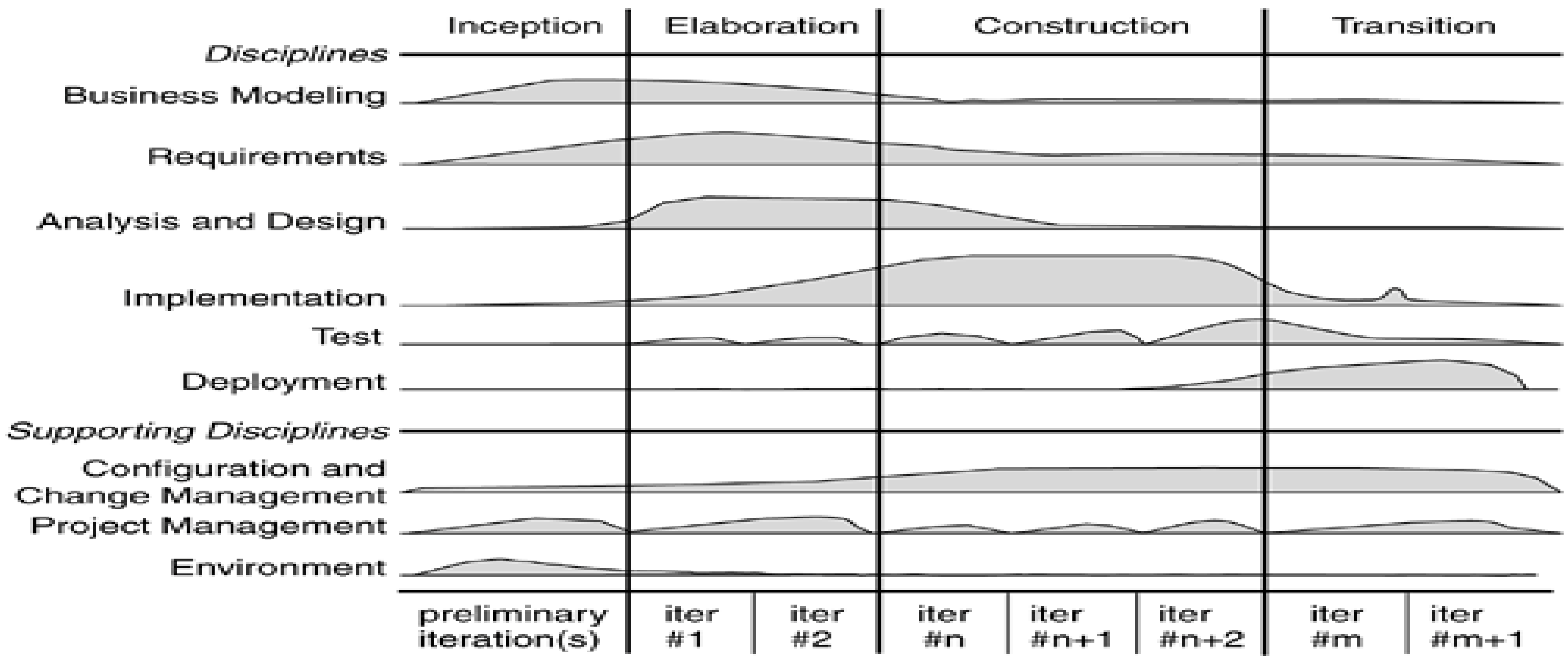
1. In Spiral, you begin by producing the objectives and needs and generating solutions for the current iteration.
2. Identify and assess risks and evaluate those solutions.
3. Move on to developing and testing the product in the current iteration.
4. Once you have a product that satisfies the objectives, you move on to planning the next iteration.



Iterative is a continuous and never-ending process until it reaches its allocated time and budget

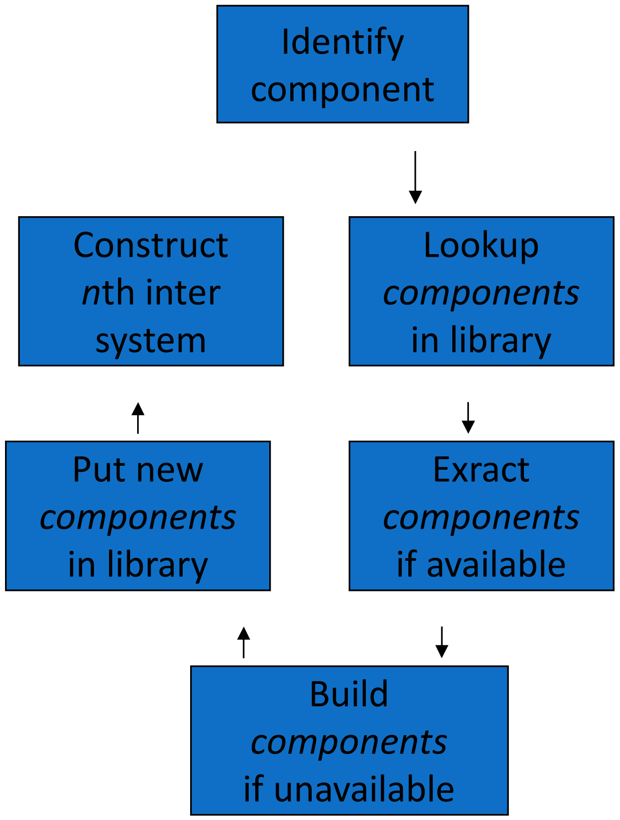
Gradually build up a product, by repeating the phase cycle until project has been completed by client SATISFACTION!

ITERATIVE: RATIONAL UNIFIED PROCESS MODEL (RUP)



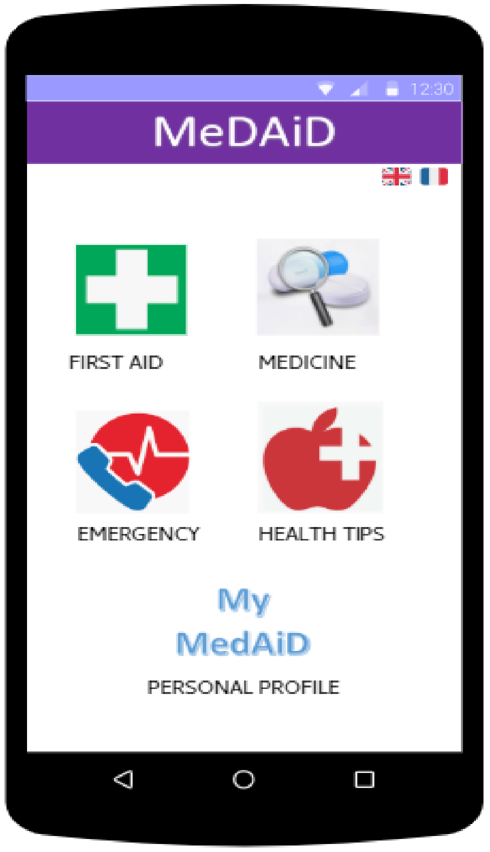
COMPONENT BASED DEVELOPMENT MODEL: SOFTWARE REUSE

- Software reuse is the concept of using developed software, commercially available software packages, and others to efficiently develop new software
- It requires standardization in the development phase, improvement of module independence (low coupling), and management of module-based units (high cohesion)
- Modules created as components have high reliability and can be expected to improve development productivity (i.e., shortening of the development period) and quality.
- The reuse of software packages may require customization.
- Suitable for re-usable object-oriented classes
- Apply characteristics of incremental and spiral development


Identify component
Lookup *components* in library
Exract *components* if available
Build *components* if unavailable
Put new *components* in library
Construct *n*th inter system

PROTOTYPING

- The prototype model is a development technique that creates a provisional prototype in a short time and is tested and evaluated by users, with specifications finalized while changes are repeated.
- System requirements ALWAYS evolve throughout a project, so process iteration protypes is useful where earlier stages are reworked is always part of the process for large systems
- Developing software products through a series of intermediate prototypes was a theme in both the Spiral and Unified Process models
- A prototype is not an actual working software product



PROTOTYPING

1. Illustrative prototype: Mock-up, and Wireframe (Non-interactive) to illustrate the idea which is can be easily disposable. Often pen and paper used to sketch the idea.
2. Throw-away prototype: Objective is to understand the system requirements. Should start with poorly understood requirements. For example, first version of the product is just built to get a better understanding of building the second version (e.g., pencil tool)
3. Exploratory prototype: Objective is to work with customers and to explore a final system from an initial outline specification. It allows the user to interact with system functionality (interactive mock-ups) and explore what is feasible (e.g., HTML, CSS)
4. Incremental prototype: When a product is built and released in increments, it is known an incremental prototype (working software). It use a triage system (the increments are build based on priorities, e.g., very important feature, important and optional features)
5. Evolutionary prototype: Develop early version of all the feature until they are fully mature in the final product. The feature evolves from initial functionalities to more mature versions.

REFERENCES

- Pressman, R.S. (2005). *Software Engineering: A Practitioner's Approach.*
- Kelly, J. C., Sherif, J. S., & Hops, J. (1992). An analysis of defect densities found during software inspections. *Journal of Systems and Software, 17*(2), 111-117.
- Bhandari, I., Halliday, M. J., Chaar, J., Chillarege, R., Jones, K., Atkinson, J. S., & Yonezawa, M. (1994). In-process improvement through defect data interpretation. *IBM Systems Journal*, *33*(1), 182-214.
- Wong, K. (2015). *Software Processes and Agile Practices*. University of Alberta