Computing Project
(COMP08053)

Lecture 4

Agile Project Management and Software Development

Traditional Project Management

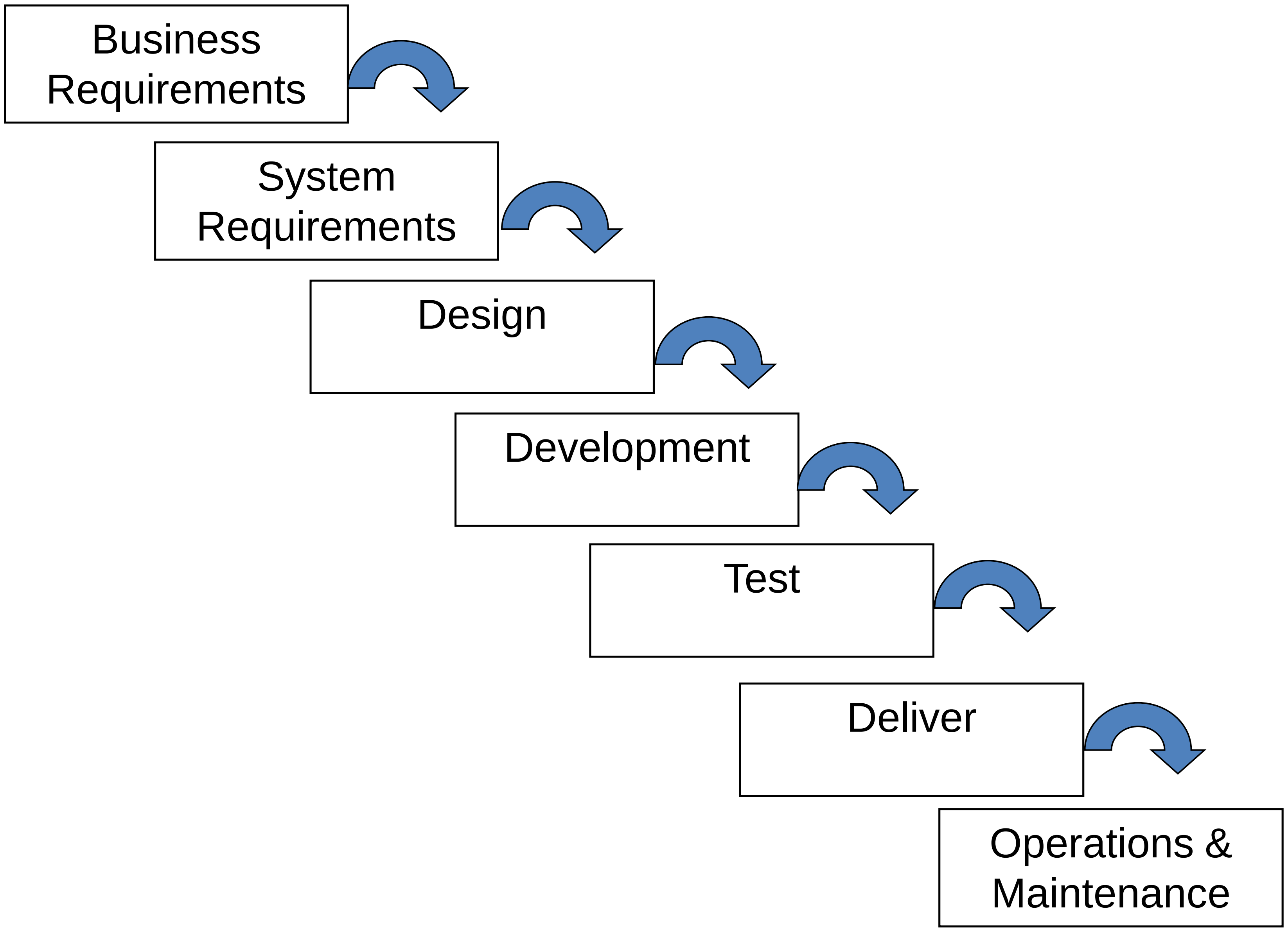
Involves a very disciplined planning and control methods

Distinct project lifecycle phases are easily recognisable

Tasks are completed in an orderly sequence, requiring a significant part of the project to be planned up front

Assumes events affecting project are predictable and activities are well understood

Waterfall Model



Traditional Project Management

Strength of approach is it lays out steps for development and stresses importance of requirements

Limitations are projects don’t always follow sequential flow and clients find difficulties in stating all requirements early in project

Traditional Project Management

Modern businesses are complex, interconnected, interdependent, interrelated and often have virtual elements

Often traditional organisational structures often do not fit

Organisations often create complex communities with strategic alliances and networks of partnerships and customers

Operate within global competition, time-to-market compression, rapidly changing technologies and increasing complexity

Agile Project Management

For project involving significant software component – traditional PM can be ineffective since requirements can be elusive, volatile and subject to change

An alternative is:

Agile Project Management (APM)

Agile Project Management

APM is a highly iterative and incremental process

Developers and project stakeholders actively work together to understand domain, identify what needs to be built and prioritise functionality

Agile Project Management

APM is useful where:

Project value is clear

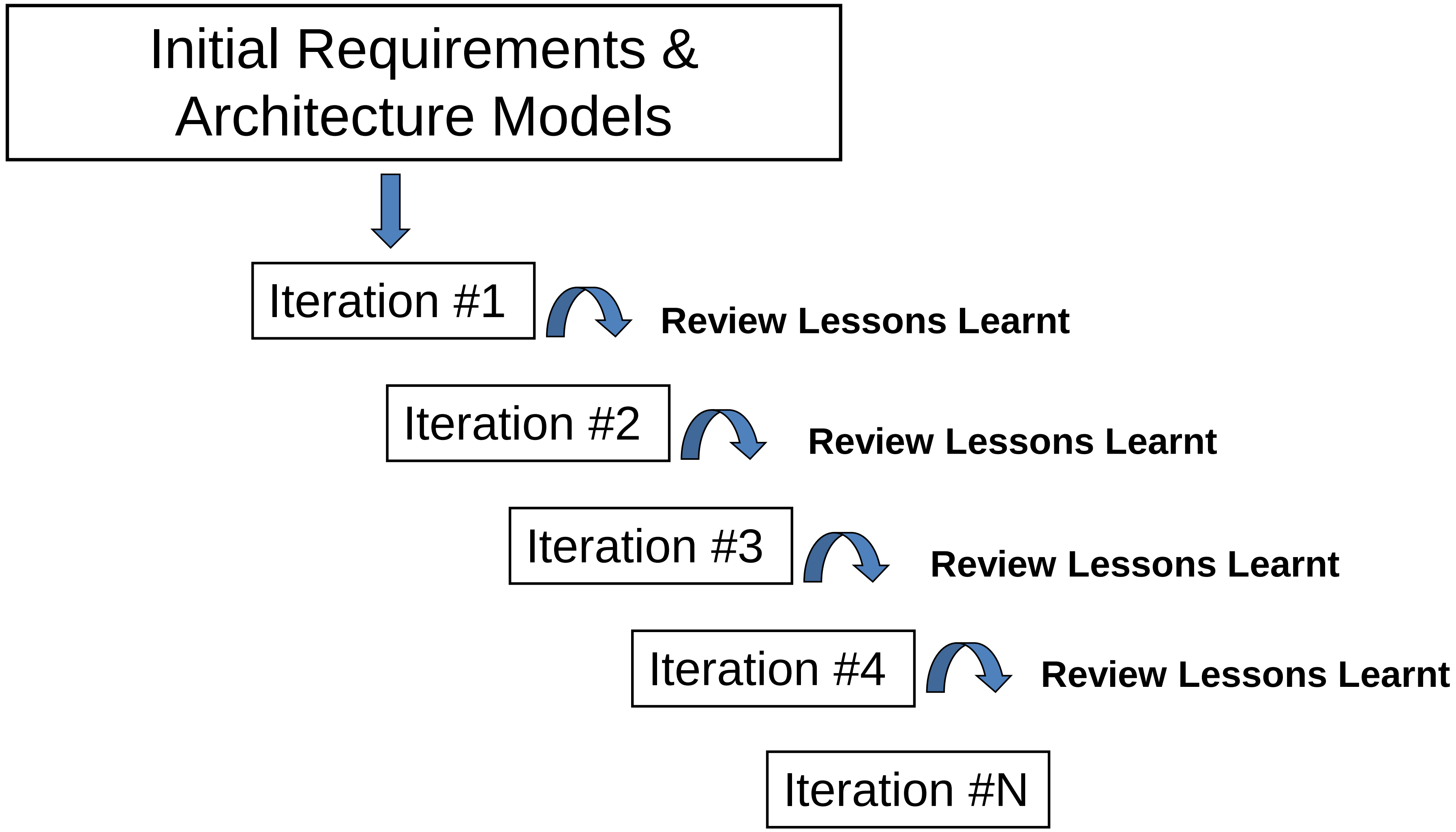
Customer actively participates throughout project

Customer, designers and developers are co-located

Incremental feature-driven development is possible

Visual documentation is acceptable (e.g. cards on the wall as opposed to formal documentation)

Agile Development Model



Agile Approach

After a streamlined planning, requirements definition and solution design phase completed – this is iterated

Activities take place in waves

Allows for immediate modifications of product as requirements come into view

Requires dedicated full-time project team including customer or end user where team work from same location

Agile Approach

Consists of many rapid iterative planning and development cycles

Allows a project team to constantly evaluate the evolving product

Allows immediate feedback to be obtained from stakeholders

Team learns and improves the product, as well as working methods from each successive cycle

Agile Approach

Whereas traditional project management emerged from engineering and construction in mid 20th century

APM conceived in 21st century by software developers for software developers to achieve better results

Agile Approach

Core team usually consists of:

- Two developers who write code in pairs for quality control
- The customer/end user
- IT architect(s)
- A business analyst
- A project manager

Agile Approach

Work is accomplished through a series of sessions where team writes code then tests working modules of system and repeats process

There is minimal documentation as team relies on informal internal communication

Agile team identifies and prioritises features based on business value

After high risk components are produced, work on highest value features first

Works if system can be delivered incrementally to customer

Agile Management Components

1. **Visual control** – ‘card on the wall’ method of planning. Team can see at a glance where they are
2. **Co-located high-performing teams** – this greatly increases quality of coordination and communication
3. **Test-driven development** – useful when customer is finding it difficult to articulate requirements. Requires more iteration
4. **Adaptive Control** – agile teams continually adapt to improve their methods as they incorporate lessons learned from previous cycle into next iteration

Agile Management Components

5. **Collaborative development** – capture candid feedback and implement lessons. Constant feedback and improvement. Constant collaboration with customer
6. **Feature-driven development** – allows team to focus on one feature at a time. High risk components built first
7. **Leadership and collaboration rather than command and control** – project manager removes barriers hindering the core Agile teams
8. **Move from Cost to Revenue** – features prioritised based on value such as increased revenue or market share

Agile Management Components

9. **Lessons learned** – after each cycle team holds a lessons learned session. How are they going to do things better on next iteration. Continually improves team performance

Just enough planning is done up-front

Customer sees and experiences a working prototype thus better able to refine or redefine requirements

Agile Approaches

Methods that are specifically designed around agile development include:

DSDM (Dynamic Systems Development Method)

Iterative waterfall mixture – popular in Europe

Strong focus on holding time and budget constant and allowing requirements to be variable in making project delivery tradeoffs

Agile Approaches

Scrum

A framework for managing projects with minimal documentation and high interactivity

Most popular agile approach

Daily scrum meetings lasts no more than 15 minutes

Each member answers 3 questions (i) What did you get done yesterday? (ii) What do you commit to today? (iii) What are your impediments?

Agile Approaches

Scrum

30 day calendar iterations

Demo to external stakeholders at end of each iteration

For each iteration, client-driven, adaptive planning

Agile Approaches

Extreme Programming (XP)

Programming method that fits well within many agile approaches

Paired programming – two developers sit together writing one piece of code

Advocates frequent releases in short development cycles (timeboxing)

Intended to improve productivity and introduce checkpoints where new customer requirements can be adopted

Software Prototyping

Most agile methods rely heavily upon prototyping techniques

Boundary between prototyping and normal system development blurred – many systems developed using evolutionary approach

Principle purpose is to help customers and developers understand requirements

Prototyping can be considered as risk reduction activity

Software Prototyping

Developers receive valuable feedback

Missing services and functionality can be detected

Allows software engineer some insight into accuracy of initial project estimates

Also insight as to whether deadlines and milestones proposed can be successfully met

Prototyping Process

1. Identify basic requirements
2. Develop initial prototype
3. Review
4. Revise and enhance prototype

Types of Prototyping

Evolutionary prototyping

Initial prototype is produced and refined through number of stages to a final system

Built in a robust and structured manner

Constantly refine it

Types of Prototyping

Throw-away prototyping

Creation of a prototype that will eventually be discarded

Used to validate or derive system requirements and discover problems, reduce risk

Can be done quickly

User interface prototyping – construct interfaces that users can test

Disadvantages of Prototyping

Developer misunderstanding of user objectives

Insufficient analysis

Developer attachment to prototype

Excessive development time of prototype

Advantages of Prototyping

Reduced time and costs

Improved and increased user involvement

Misunderstandings exposed

Improved system usability

Reduced overall development effort