Methods and Frameworks – an introduction

Requirements Management (COMP1786)

The need for a framework

- Purpose of the systems development function is to develop **effective** systems in the most **efficient** way possible.

- **The traditional systems development life cycle** (with which you are all familiar) was a first attempt to provide a controlled environment for systems development.

What we need from a systems development methodology

- It is an attempt to specify in great detail:
  - the generic development framework
  - those steps essential to the development process
  - those steps considered optional
  - the order in which the steps should be performed
  - the tools and techniques required in order to undertake these steps
  - The end-deliverables that we need

Features of a generic system development methodology

- **The Technical model**
  - Defines the tools and techniques that will be required in order to follow the methodology framework.
  - It will consist of the use of a combination of the information systems modelling views: the data model, the process model, the behavioural (or dynamic) model.

Cont..

- It may also consist of methodology-specific tools & techniques.
- Also computer-based tools to assist with the development of the information system. (CASE, IPSE packages)

Cont……

- **Managerial model**
  - This part of the methodology covers the order in which steps are to be covered, how to control the activities.
  - It provides the framework within which the development will take place.
  - It defines stages, steps, tasks to be undertaken.

Cont..

- Most importantly, it defines:
  - **When** they will be undertaken
  - **Which** are critical/ which are not
  - **How** they should be managed
  - **How** developers and teams should work together
  - **What** the **end-deliverables** will be.

Where we've got to today..

- Development techniques have changed rapidly over the last 15 years
- The approach to systems development still continues to evolve as we utilise new technology and s/w development approaches
  - Development approaches for web-based projects are poor
  - Many systems developed using system building tools

There is some agreement

- All s/w developers agree that in the end the important issue is not so much about which methodology you use, as having *some kind of framework within which to structure your project.*
- Today's developers have a much more realistic view: *you need to adapt your recipe to the ingredients you have and to the preferences of the consumers*

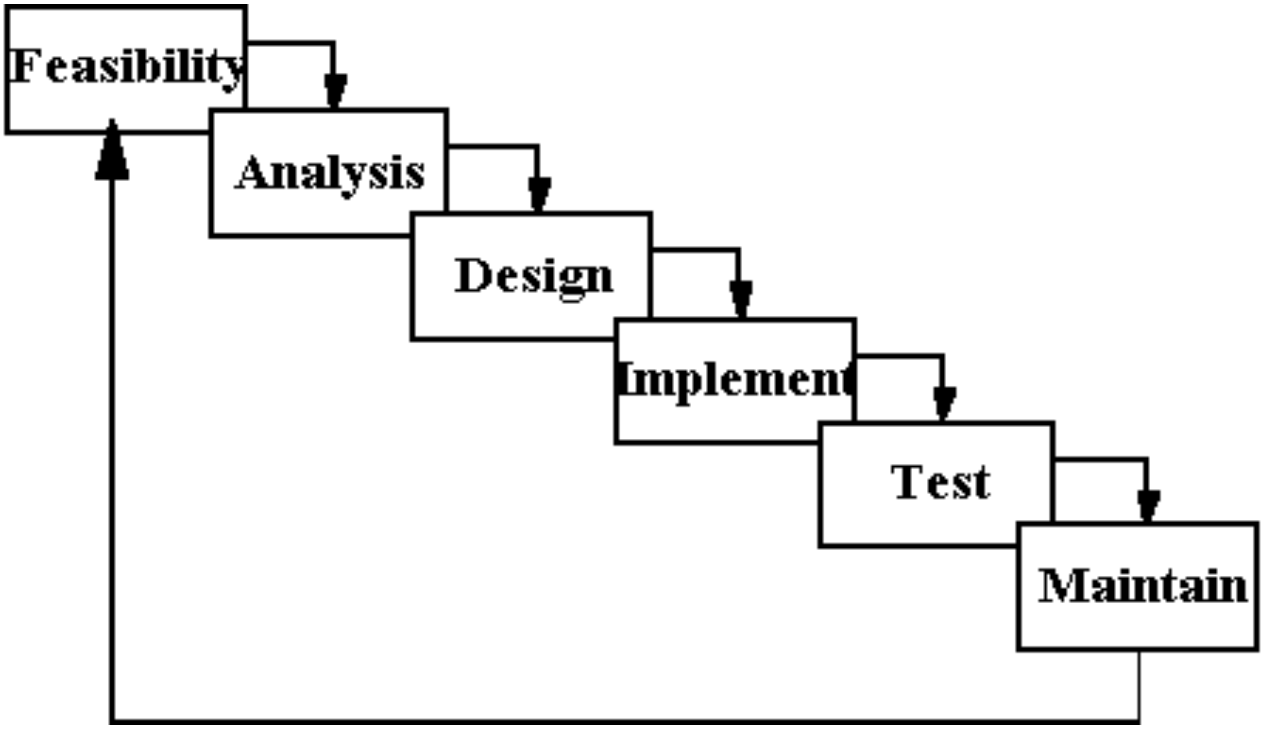
The remainder of the lecture

- Significant developments in methods and frameworks over the years
  - SSADM
  - OMT, OOAD
  - PRINCE
  - RUP
    - UML (not a framework/method!)
  - Agile
    - XP, Scrum, DSDM, Agile PM, KANBAN, SAFE
- Consider how relevant they are

Systems development life cycle

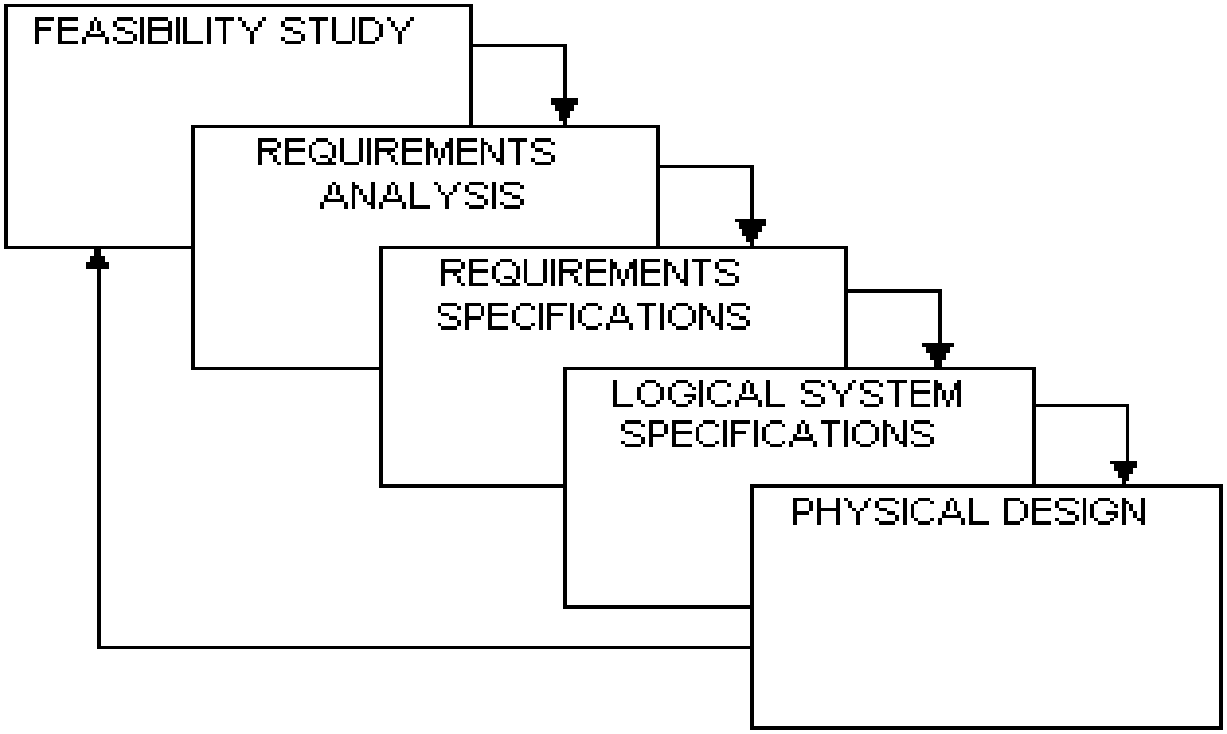
- The late 1960's saw the development of a number of *methods* and *frameworks* designed to overcome the problems associated with systems dev. projects.
- Outlined the key activities forming the SDLC.
- Main criticisms: rigid, no back-tracking or iterations allowed
- Modern dev. frameworks built from this.

You must remember this!



SSADM (1)

(Structured systems analysis & design method – 1980s)



Looks at Analysis and Design phases of the waterfall method

SSADM (2)

- Structure Analysis and Design Method
- Developed by Learmonth and Burchett Consultancy (LBMS)
- Adopted for public sector use by government in 1983
- Originally aimed at trad. waterfall development lifecycles
- Gained a reputation for being bureaucratic, unwieldy, unproductive

SSADM (3)

- In 1990s different versions of it appeared to try and make it more attractive to developers
  - Gave it a ‘cook book’ approach
  - Rapid Application Development version
  - Object Oriented version
- These days considered ‘ an old fashioned approach’

SSADM (4) – Very rigid

- 7 stages
  - Feasibility study
  - Investigation of current environment
  - Business system options
  - Requirement specification
  - Technical systems options
  - Logical design
  - Physical design
- Output from one stage is input to next
- Lead some projects into "analysis paralysis“

SSADM (5) stages

- Each stage has a number of steps
- Each step may have a number of substeps
- Each stage/step/substep has one or more end deliverables associated with it…..

Example: SSADM
Requirements Analysis Stage

- Investigation of current environment
  - Establish analysis framework
  - Investigate and define requirements
  - Investigate current processing
  - Investigate current data
  - Derive logical view of current services
  - Assemble investigation results
- Business System options
  - Define business system options
  - Select business system options
  - Define Requirements

Is it worth bothering with?

- Have to remember that it has left a legacy in most public sector systems as SSADM
- Also left a legacy in many large organisations as LBMS
- So, many systems still in existence that were developed in these environments
- www.itjobswatch.co.uk ranks it 1134 (up 54 points on last year, at which time it was up 21 points on the previous year)
- Around £61000 (last year it was £55000, and it was £40000 4 years ago in 2014)

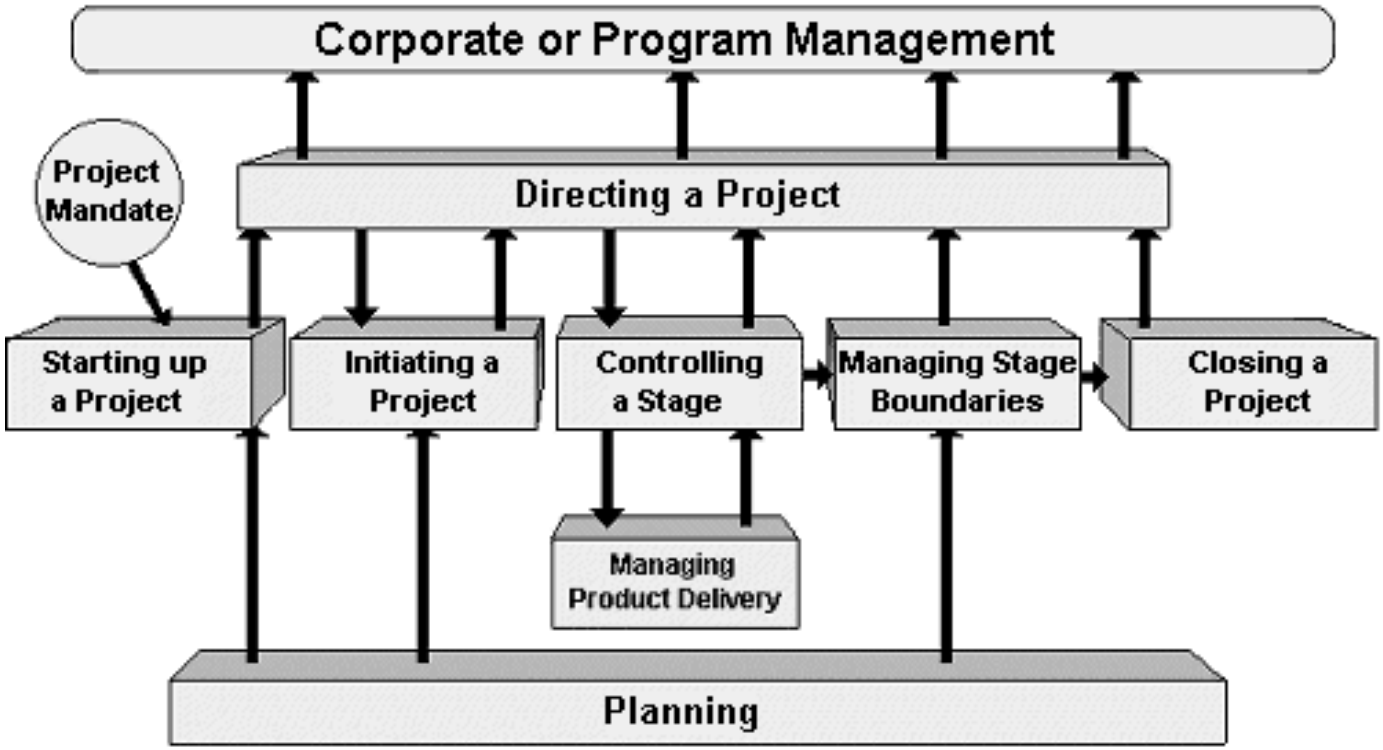
Prince (1)

- Project in controlled environments
- Not a systems development methodology
  - There is often confusion about this
  - It is a Project Management Framework
- Aimed specifically at project organisation, management and control.
  - Not just for IT based projects
- History back to 1989 – CCTA (government agency)

Prince (2)

- Now UK's standard for project management in the **Public Sector**!
- Prince2 is the latest version
- Pay big money to attend courses/ get certification

PRINCE2 framework



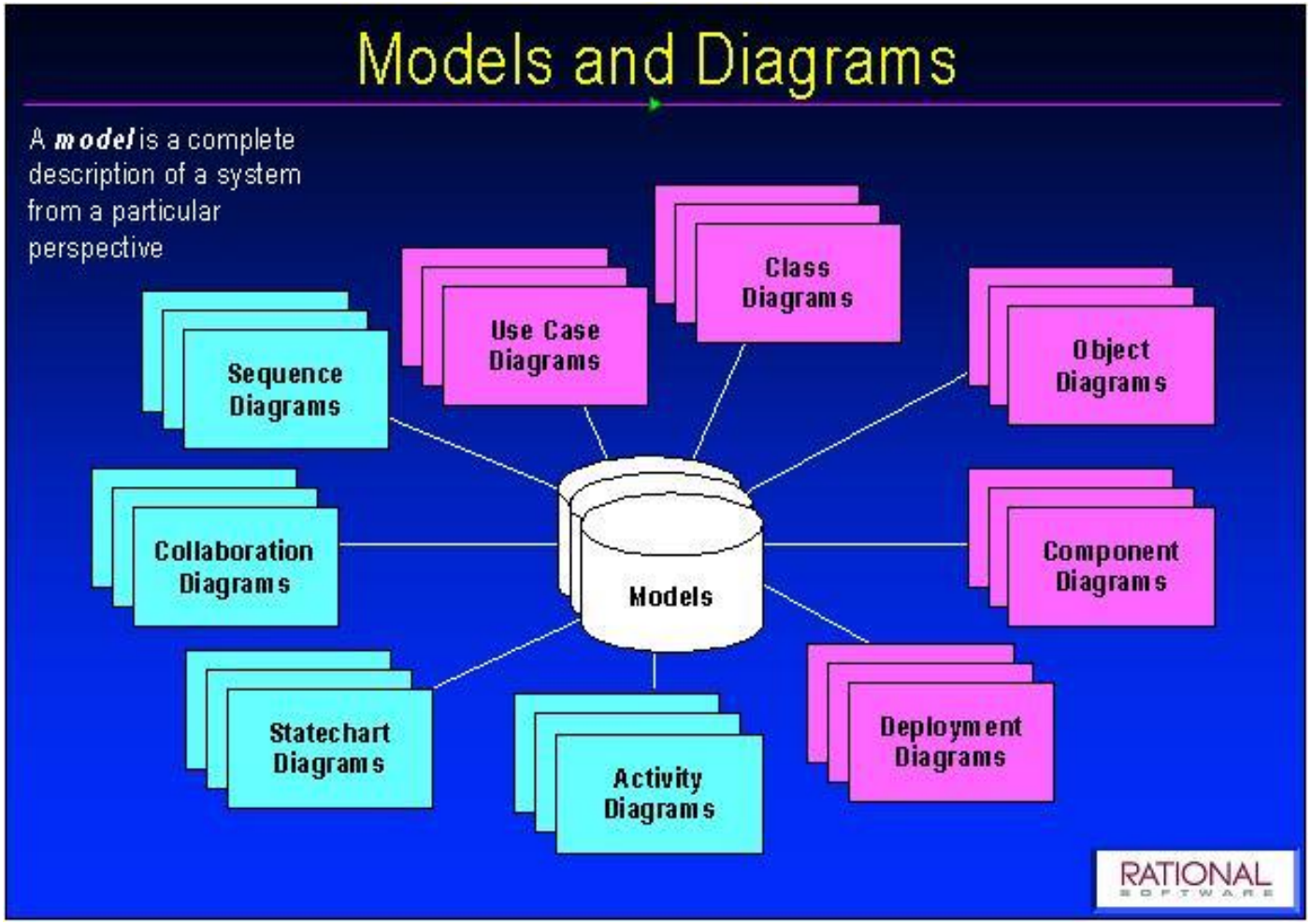
Is it worth bothering about?

- www.itjobswatch.co.uk ranks it 154th this quarter (down 28 points on last year, at which time it was down 15 points on the previous year and 35 the year prior)
- Average salary £55,000 (£2500 up from last year)
- Need to have Prince certification (costs of training)
- http://www.prince2.com/whatisp2.html

UML

- Unified Modelling Language
- Again, not a methodology based upon our definition
  - A set of specifications and design notations for object oriented development
  - Developed combining modelling techniques developed by Booch, Rumbaugh and Jakobsen
- Lots of methods have been developed from it.

UML models and diagrams



Is it worth bothering with?

- Yes, it is standard for OO modelling
- Good thing is that certain models (eg use cases) can be used outside of OO modelling world
- Big criticism, though, is that it is too complex
- www.itjobswatch.co.uk rank it at 374 (down 52 points, which was down 23 points on the previous year)
- Average salary around £52500 (down £2500 from last year, the same as 2016)

RUP
The *Rational Unified Process*

- “A full-fledged process able to support the entire software development life-cycle”.

- “Use case driven, architecture centric, iterative and incremental”

Use case driven

- Use cases used here to capture *the user requirements*
- A use case describes an *element of the functionality of the system*
- All the use cases together (the use case model) depicts the total functionality of the system
  - The system is designed based upon the use cases
  - Implemented to support the use cases
  - Tested upon content of the use cases
- *So, Use cases are very important!*

Architecture centric

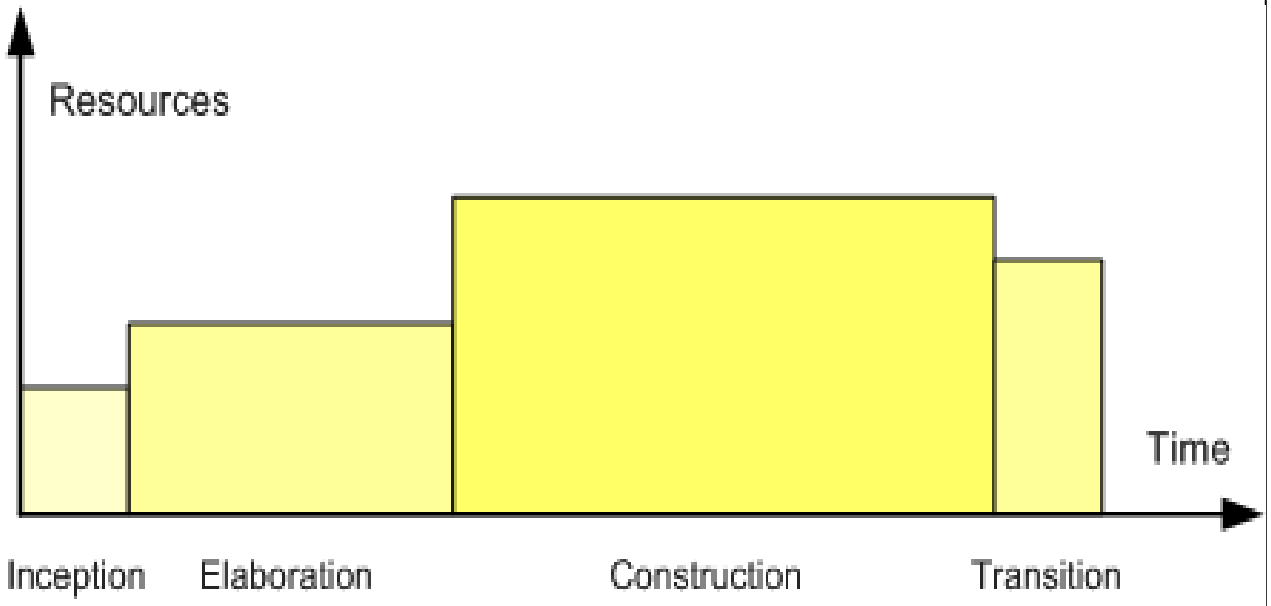
- End deliverables or ‘artefacts’ are very important in the RUP just as they are in UML.
- Compares s/w architecture to architecture of a house.
- Many methods ignore devt of the architecture (models etc) and concentrate on devt of software.

Iterative and Incremental (1)

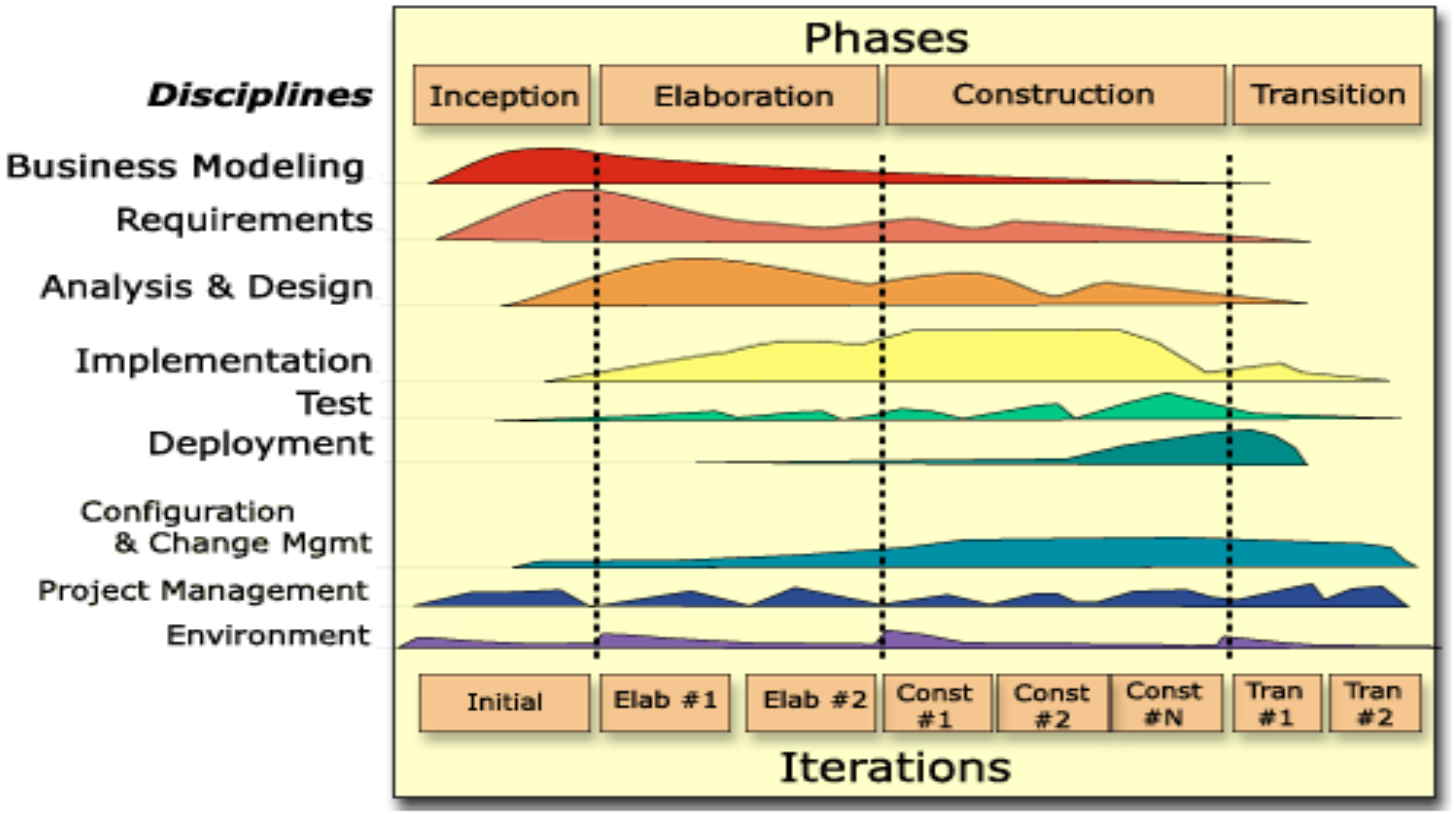
- Belief that user requirements cannot be fully and accurately defined at one go.
- Requirements evolve and change over time as understanding deepens.

- *This is a key concept of modern methods and we will come back to it over and over again.*

RUP – Phases and resources



RUP – process structure



Is it worth bothering with?

- Yes, to an extent
  - Still used on large scale products
  - Owned by IBM
  - www.Itjobswatch.co.uk rank it at 1004 (down 146, which was down 180 on the previous year)
  - Average salary around £65000 (up £7500 from last year)

Agile Methods (1)

- Agile methodologies are a family of methodologies, not a single approach to software development.
- They are based on the original concepts of Rapid Application Development (RAD) – more in coming weeks
- The idea has been developed to incorporate later stages of the life cycle (software development) and project management
- DSDM is an Agile Method developed in the UK

Agile Methods (2)

- In 2001, 17 prominent figures in the field of agile development (then called "light-weight methodologies") came together to discuss the unifying theme of their methodologies.
  - Formed the Agile Alliance
  - Created the Agile Manifesto
- The background of these ‘experts’ was in software development rather than in requirements definitions/business analysis and so a new way of thinking for this discipline

Agile Methods (3)

- The Agile Manifesto
  - We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
    - **Individuals and interactions** over processes and tools
    - **Working software** over comprehensive documentation
    - **Customer collaboration** over contract negotiation
    - **Responding to change** over following a plan
    - http://agilemanifesto.org/

Agile Methods (4)

- Some of the well-known agile software development methodologies include:
  - Extreme Programming (XP)
  - Scrum
  - Adaptive Software Development (ASD)
  - Crystal Clear and Other Crystal Methodologies
  - DSDM
  - Feature Driven Development
  - Lean software development
  - KANBAN
  - Agile PM

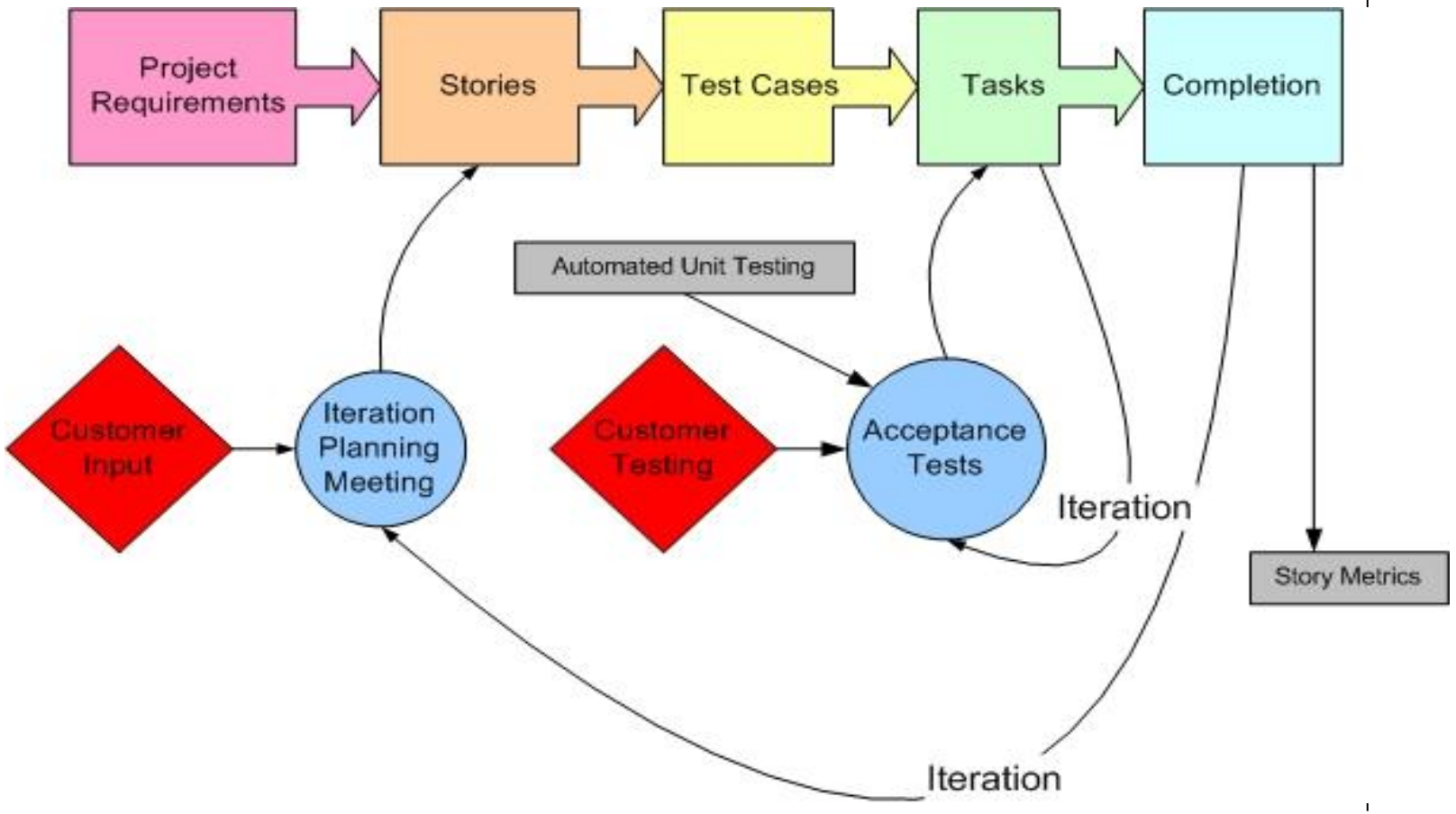
Agile Methods (5)

- Key Agile concepts
  - Most agile methods attempt to minimize risk by developing software in short timeboxes, called iterations, which typically last one to four weeks.
  - Each iteration is like a miniature software project of its own, and includes all the tasks necessary to release the mini-increment of new functionality: planning, requirements.

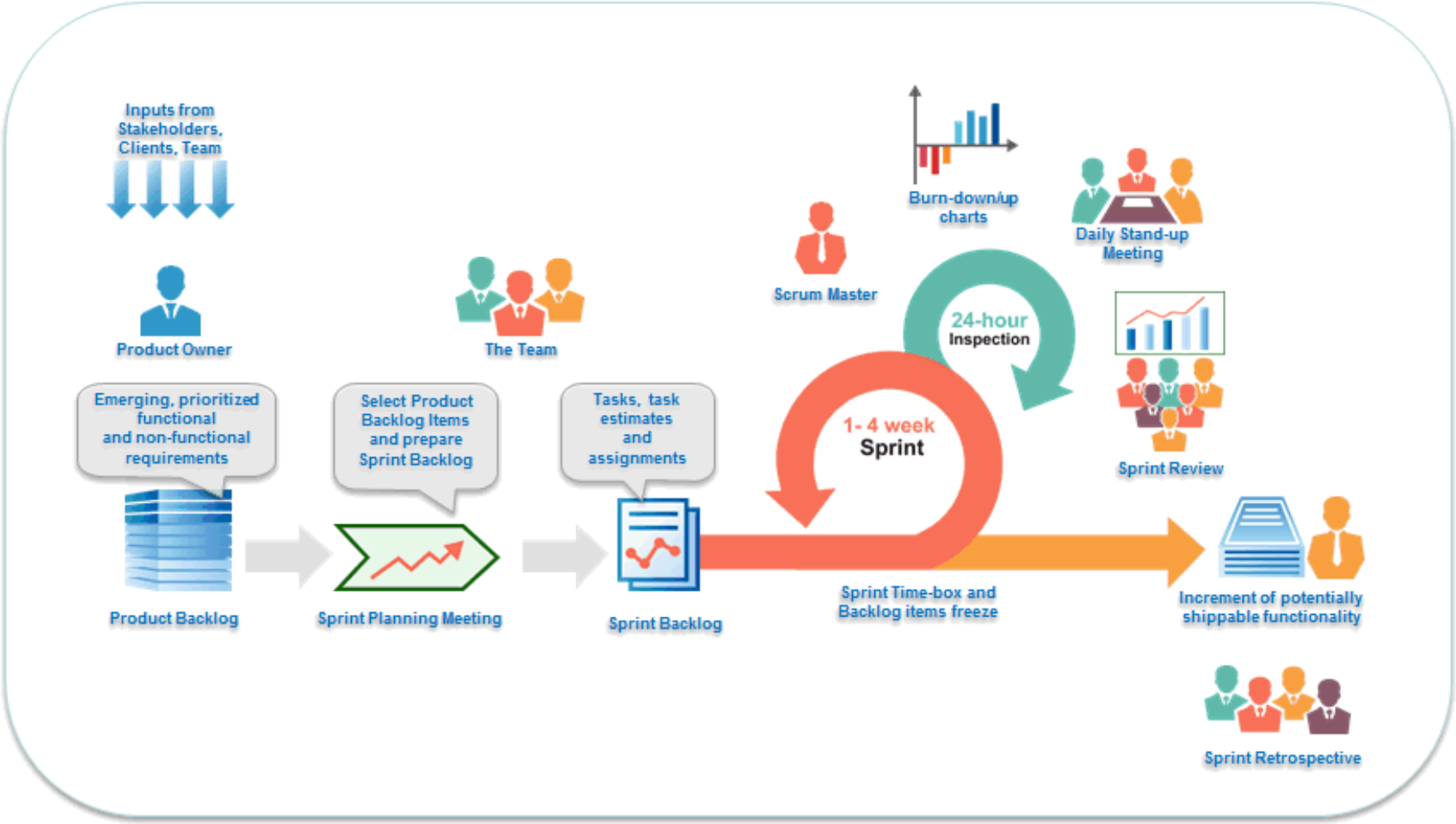
Agile Methods (6)

- Key Agile concepts cont…
  - Agile methods emphasize face-to-face communication over written documents.
  - Most agile teams are located in a ‘bullpe’n and include all the people necessary to finish software.
    - At a minimum, this includes programmers and their "customers."
    - The bullpen may also include testers, interaction designers, technical writers, and managers.

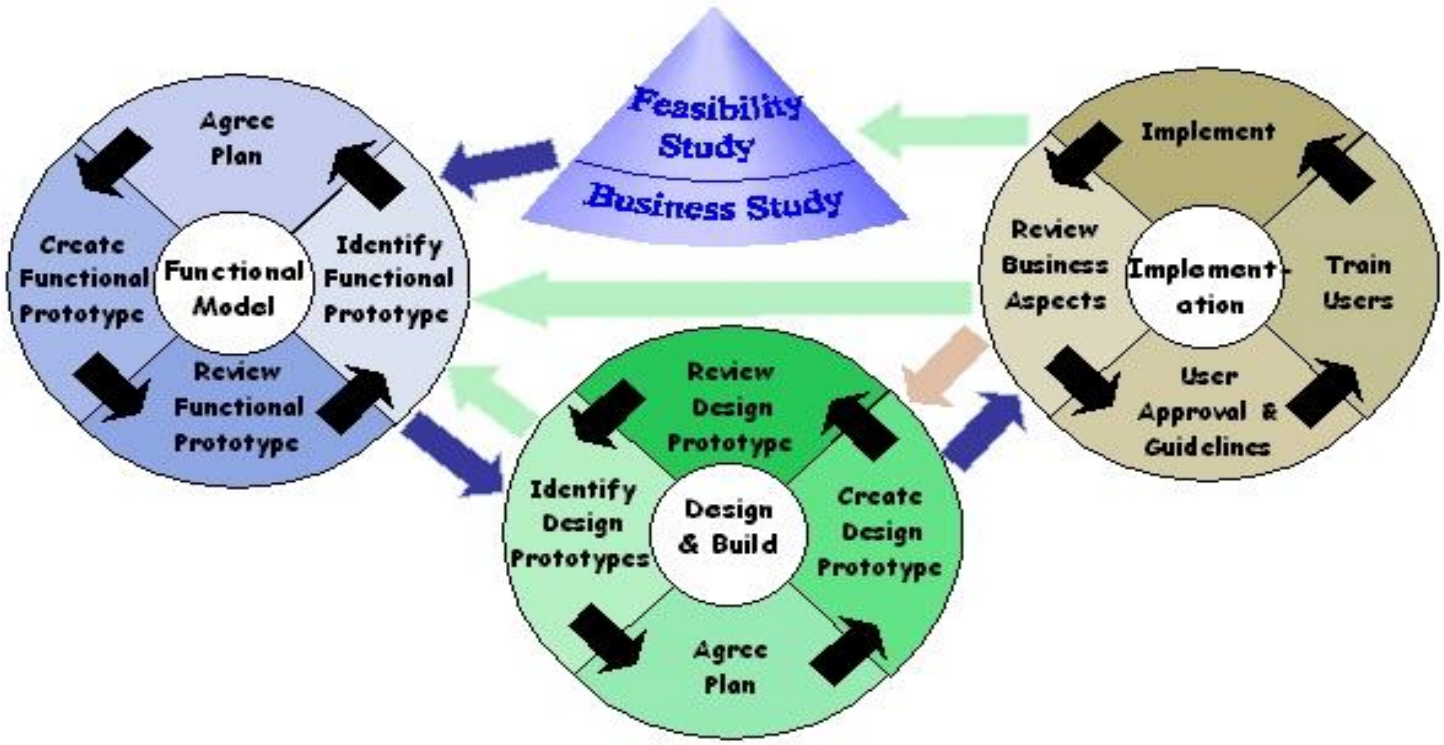
Extreme Programming



Scrum


Inputs from Stakeholders, Clients, Team
Product Owner
The Team
Scrum Master
Burn-down/up charts
Daily Stand-up Meeting
24-hour Inspection
Sprint Review
Emerging, prioritized functional and non-functional requirements
Select Product Backlog Items and prepare Sprint Backlog
Tasks, task estimates and assignments
1- 4 week Sprint
Product Backlog
Sprint Planning Meeting
Sprint Backlog
Sprint Time-box and Backlog items freeze
Increment of potentially shippable functionality
Sprint Retrospective

DSDM


Feasibility Study
Business Study
Agree Plan
Identify Functional Prototype
Review Functional Prototype
Create Functional Prototype
Functional Model
Review Design Prototype
Create Design Prototype
Agree Plan
Identify Design Prototypes
Design & Build
Implement
Train Users
User Approval & Guidelines
Review Business Aspects
Implement-ation

Agile development

Agile Software Development

- Ranked 1st (same as last year, after ranking 2nd for three consecutive years. Was at 4 in 2014)
- Average salary around £55000 (same as last year, which was up £2500 from the previous year, and another £2500 from the year prior)

Remember, there are many different Agile development frameworks!

Summary

- Many different methods/frameworks exist
  - Make sure you know the difference between a development method/framework, a project management method and a Modelling framework (such as UML)
- Original structured methods such as SSADM are no longer used (too rigid) but legacy systems that used them still exist

Summary (2)

- Methods such as RUP designed for larger scale systems are still used but they are much more flexible
- Agile (RAD) methods are currently in vogue – still criticism of them
- Current approaches advocate
  - Time boxes, working in small groups, integrating analysts, designers, programmers and users