



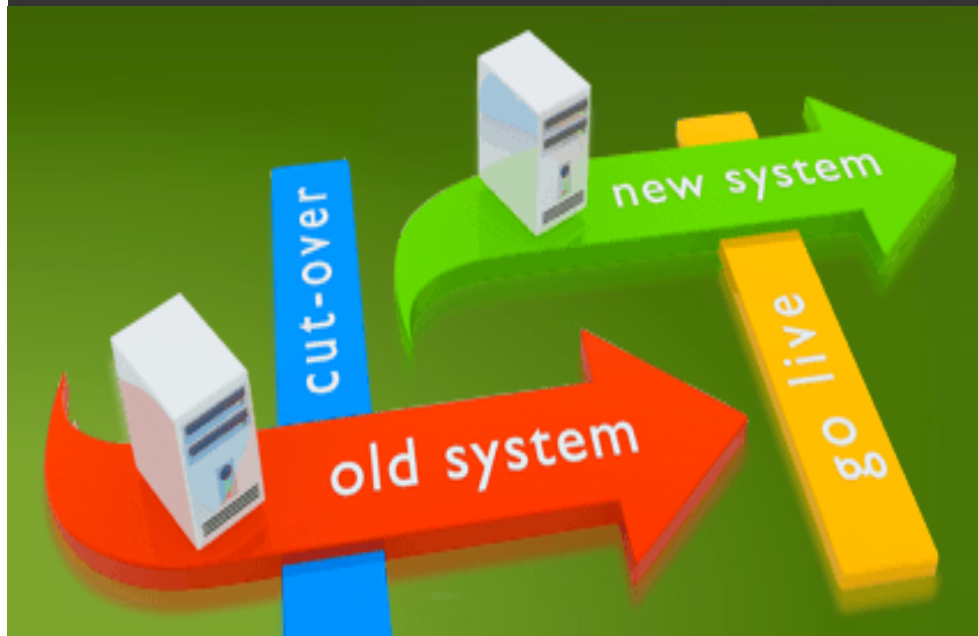
MONASH University

Information Technology

# FIT2001 – Systems Development

## S11 - Implementing & Maintaining the System

Chris Gonsalvez



# Our road map

- What are Information Systems?
- How do we develop them? Systems Development (SDLC) – key phases
- Traditional vs. Agile approaches to developing systems
- Some System Development roles and skills
- Understand the requirements gathering process
- Managing stakeholders
- Requirements gathering and documentation techniques
- Prototyping & Interface Design
- Detailed Design – Design Class Diagrams & Sequence Diagrams
- Testing

- Implementation
- Build/Buy, Outsourcing
- Maintenance / Support

# At the end of this seminar you will:

- Understand the range of activities in the Implementation phase of the SDLC;
- Understand the advantages and limitations associated with building vs. buying IT applications
- Appreciate the reasons for which many organisations outsource all or parts of their IT function, and recognise the benefits and concerns
- Be aware of the significance of support activity in the work of an IT department and, understand the different types of support activities.

# Lecture Outline

## Implementing the System – Tasks

1. Implementation Planning
2. Build vs. Buy the System / Outsourcing
3. Testing (covered in Lecture 9)
4. Documentation
5. Get the system ready for production
  - 5.1 Data Conversion / Migration
  - 5.2 Training
6. Deploy the system
7. Wrap up / Transition to support phase
  - 7.1 Maintenance
  - 7.2 Change management
  - 7.3 System closure / Post Implementation review

# Implementing the System.1

1. Implementation Planning
  - *Review Acceptance Checklist, Prepare Implementation Schedule*
2. Build the System or Buy the System
  - *Will result in a varied implementation path*
3. Test the System (covered in Week 10)
  - *System Testing (functional and performance), Acceptance Testing*
4. Finalise documentation
  - *System documentation, User documentation*

## Implementing the System.2

5. Get ready for the System to go into production
  - *Data conversion / migration, Configure the production environment, Conduct training*
6. Deploy the system
  - *Install / deploy the system, Monitor operations, Benchmark testing, Tune the system,*
7. Wrap up
  - *Operations handover, Transition to Support, System Closure, Post-implementation review*

*Note: Implementation tasks are carried out throughout the development process*

# Implementation Planning

### ■ Implementation Plan

- *Developed early in SDLC for Waterfall, at the start of each sprint for Agile or varies depending of when product is going to be released*
- *Needs to be constantly reviewed/updated to reflect changes during development*
- *Requires a great deal of co-ordination - involves many professionals external to the IT development team*
- *Poor planning can cause significant delays to the deadline*
  - *Data not ready*
  - *Training not completed*



## Information Technology

# Build vs. Buy





# Build vs. Buy

### BUILD

Custom build IT applications in-house

### BUY

Purchase or lease standard IT applications that are commercially available

- may range from simple to very complex systems

### BUY - Useful ...

- When you need an IT application for a generic company function eg. payroll
- When resources for in-house development are in short supply
- When the BUY option is more cost effective than in-house development
- Usually most of the design and implementation tasks are done so there is significant time saving
- Because the system and documentation are usually maintained by the vendor
- Because the design specification is fixed, so no reworking and users have to accept it
- Politically because:
  - external work is often perceived as being superior to an in-house effort so its easier to get a new systems into the company
  - easier to get management support because of fixed costs
  - problems can be attributed to the package rather than internal sources so cuts out a source of internal conflict

### BUY – Limitations ....

- Very rare to find a package that can do everything well that a user wants
- Often need to develop specialised package additions because the multi-purpose packages do not handle certain functions well
- Conversion and integration costs can sometimes be so significant as to render the project infeasible
- Some vendors refuse to support packages which have been customised by the users .. and most packages need some customisation
- Customisation can be so extensive that it would have been cheaper to develop the system in-house

### **Build vs. Buy? Further considerations**

- Functionality
- Cost
- Vendor Support
- Viability of Vendor
- Widespread use by others(client references)
- Flexibility
- Documentation
- Performance and scalability
- Ease of Installation



# IT Outsourcing



# IT Outsourcing – What?

The practice of turning over some or all of an organisation's IT applications and/or operations to an external provider

- Involves a contractual agreement that involves the exchange of services for payment
- Often includes the transfer of assets (eg staff)

According to Forrester Research:

- Overall investment in tech outsourcing and hardware maintenance in 2019 estimated to be \$703 billion

# IT Outsourcing – Why?      Benefits

- Reduced costs
- Strategic – focus on core business activities
- Access – to latest skills and technologies

.... favourable market reaction  
– increase in share price

# Outsourcing – Need to consider:

- Which aspects of the business will be outsourced – value proposition
- Effect on the business
- Cost to get these services delivered
- Level of required service
- Commitment of the stakeholders
- Board, senior management, staff
- How the relationship be governed - contracts
- Transition
- Selection of external organisation



# The Outsourcing process - Phases

**The six phases of activity can be defined as follows:**

**Assess:** Define objectives and assess capacity.

**Prepare:** Service level definition and RFP creation.

**Evaluate:** Response evaluation and supplier selection.

**Commit:** Contract development and finalisation.

**Transition and Transformation:** New service implementation.

**Optimise:** On-going supplier and vendor management.

# Testing

Covered in Lecture 10 last week

# System Documentation

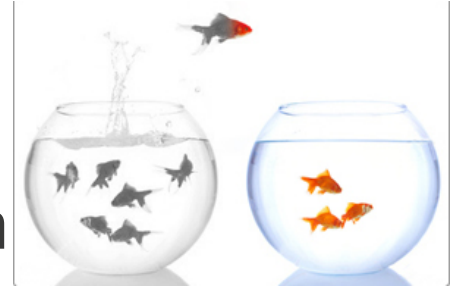
- Used to facilitate communication during development
  - *Generated as a by-product of development*
- Required for the day-to-day running, maintenance and enhancement of the system after development
  - *Includes, descriptions of system functions, architecture, and construction details, source code, analysis and design models*
  - *Instructions to run the system for IT Operations staff*
- Documentation has to be kept up-to-date to be useful
  - *Automated tools help*
    - *automatic synchronisation when design models used to generate software*
    - *reverse engineering used to update design models when software is updated*

# User Documentation

- *Description of how to use the system for end users*
- *Examples include User Manual, On-line help, Quick Reference guides*
- *The User Manual typically includes (ITIL User Manual template) :*
  - Introduction
  - Overview: A description of the applications and its functions
  - Setup: How to install and configure the application
  - Getting started: Instructions about how to start using the applications
  - Advance procedures: A comprehensive navigation throughout the functions in the application and how to perform advanced tasks
  - Reporting: How to execute and customize the reports available
  - Troubleshooting: Recommendations for solving most common issues
  - References: A comprehensive list of error codes
  - Appendices: A glossary, a list of tables, a list of figures and an index

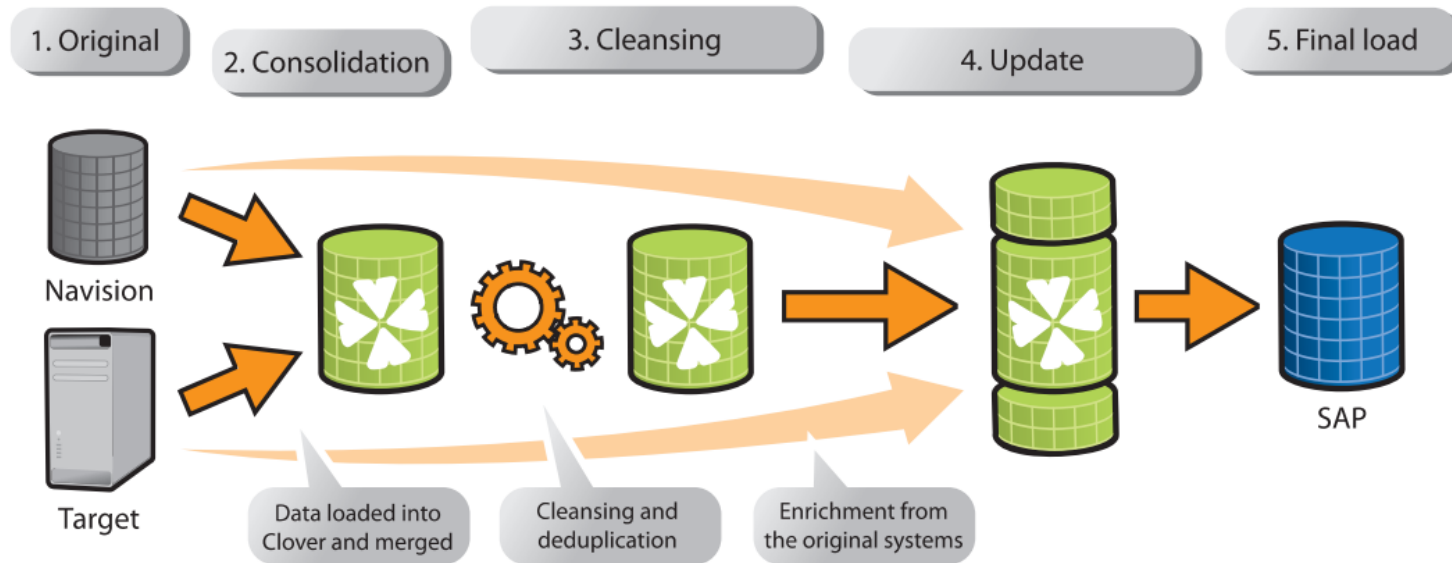


# Data conversion / migration



- Process of getting data ready for the new system
  - *A critical, challenging task, that can be very complex and costly*
  - *Can create new or reuse existing data*
  - *Data typically obtained from:*
    - Files or databases of system being replaced
    - Manual records
    - Files or databases of other systems being integrated with new system
    - Interaction with the client during normal system operation
  - *Need to consider:*
    - Changes in data storage format and content
    - Conversion process
      - Often needs specially written conversion programs
      - Manual file conversion is a time-consuming task
      - Confirmation of data accuracy

# Data conversion / migration



- *Prepare existing files ... no errors, up-to-date*
- *Prepare manual files*
- *Add new data, Cleanse data, Remove duplicates*
- *Build new files and validate*
- *Begin maintenance of new and old files*
  - *can introduce time lag*
  - *files may be out of step*
- *Work towards established cut-off date*
- *Final check of accuracy*

# Configure the production environment

- Ensure all facilities are set up:
  - *If infrastructure exists – Is it suitable?*
  - *May require new hardware and system software infrastructure*
    - *must be acquired, installed and configured*
    - *must address any version changes between purchase and delivery*
  - *Consider issues such as:*
    - *adequate space for all resources, ergonomic furniture, noise reduction, privacy, security, appropriate electrical connections, uninterrupted power, etc.*
- Infrastructure requires periodic updates:
  - *Software maintenance releases*
  - *Software version upgrades*
  - *Declining system performance*

# Conduct Training

“If you think education is expensive  
and time-consuming - try ignorance”

Bok



# Conduct Training

- Considerations:
  - *Users*
    - *Number of users, Existing skill levels, On-going usage levels – regular, occasional*
  - *Level of detail to be imparted to the audience*
  - *Who should conduct the training*
  - *Where / When should the training be conducted*
  - *Methods and resources, specialised training documentation – designed to put novice users at ease*
  - *Need supportive User Manager who is committed to allocating time for training*



# Conduct Training

- Training aids
  - *Must be easy to use*
  - *Reliable*
  - *Demonstrations and classes*
  - *Training documentation*
    - *especially designed to put the novice user at ease*
  - *On-line help*
  - *Expert users*
- On-going training needs after installation:
  - *Online help*
  - *Resident experts*
  - *Help desk*

# IT Training example

**From:** FITGlobalEmails <fitglobalemails@monash.edu>  
**Subject:** [ FIT Announce ] Dedicated Faculty BI training - 15th June 2011  
**Date:** 18 May 2011 12:26:06 PM AEST  
**To:** All Faculty of Information Technology Staff <allfit@infotech.monash.edu.au>

### Faculty of Information Technology

***This notice was approved for distribution on behalf of Michelle Ketchen, School Manager, Caulfield School of IT***

Hi everyone,

The following training has now been confirmed for Wednesday 15th June at 10.30 am in H6.90. The session will run for 90 minutes. Please let me know if you would like to attend. If you have already advised me then don't worry.

**Course Details:**

The course details report provides timely access to course performance information. Course measures such as student course average mark, student course progress rate, student course retention rate, course enrolments and load (EFTSL) are available.

This is at a detailed course level and is viewable across attributes including student course details, student demographics (i.e. country, citizenship type, language, etc.), student academic history (i.e. secondary education, tertiary education, credit etc.), and a student's Monash pathway, that is, if they have completed a 'previous' course at Monash prior to enrolling in their course, if they enrolled into a 'next' Monash course, or if they transferred in or out of a course

# Install / deploy the system

- Method of installation / deployment depends on several criteria:
  - **Cost:** *if there are cost constraints certain choices are not viable*
  - **System criticality:** *if system failure would be disastrous, the safest approach should be selected regardless of cost*
  - **Disruption to business:** *what level of disruption to the company and IS operations is acceptable*
  - **User computer experience:** *the more experience the users have, the less necessary it is to delay changeover*
  - **System complexity:** *the more complex the system, the greater the chance of flaws ... a safer approach is better*
  - **User resistance:** *need to consider what the users are best able to cope with*

# Installation / Deployment alternatives

### Decision 1: *Multiple locations?*

1. Pilot/Single Location installation
2. All Locations

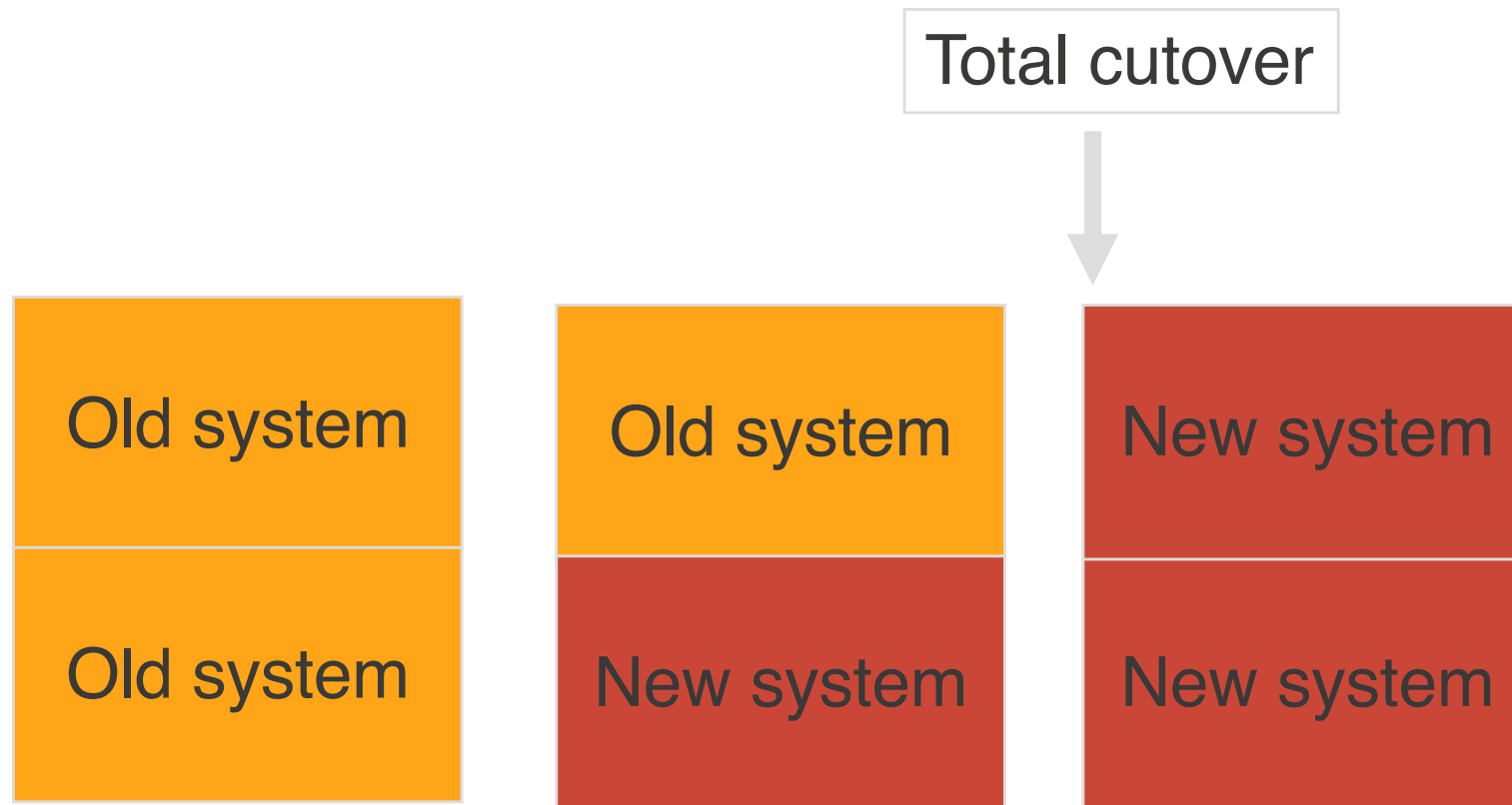
### Decision 2: *Multiple functions?*

1. Phased/Staged functionality installation
2. All functionality

### Decision 3: *Installation alternatives*

1. Direct/Abrupt installation
2. Parallel installation
3. Phased installation or Staged installation

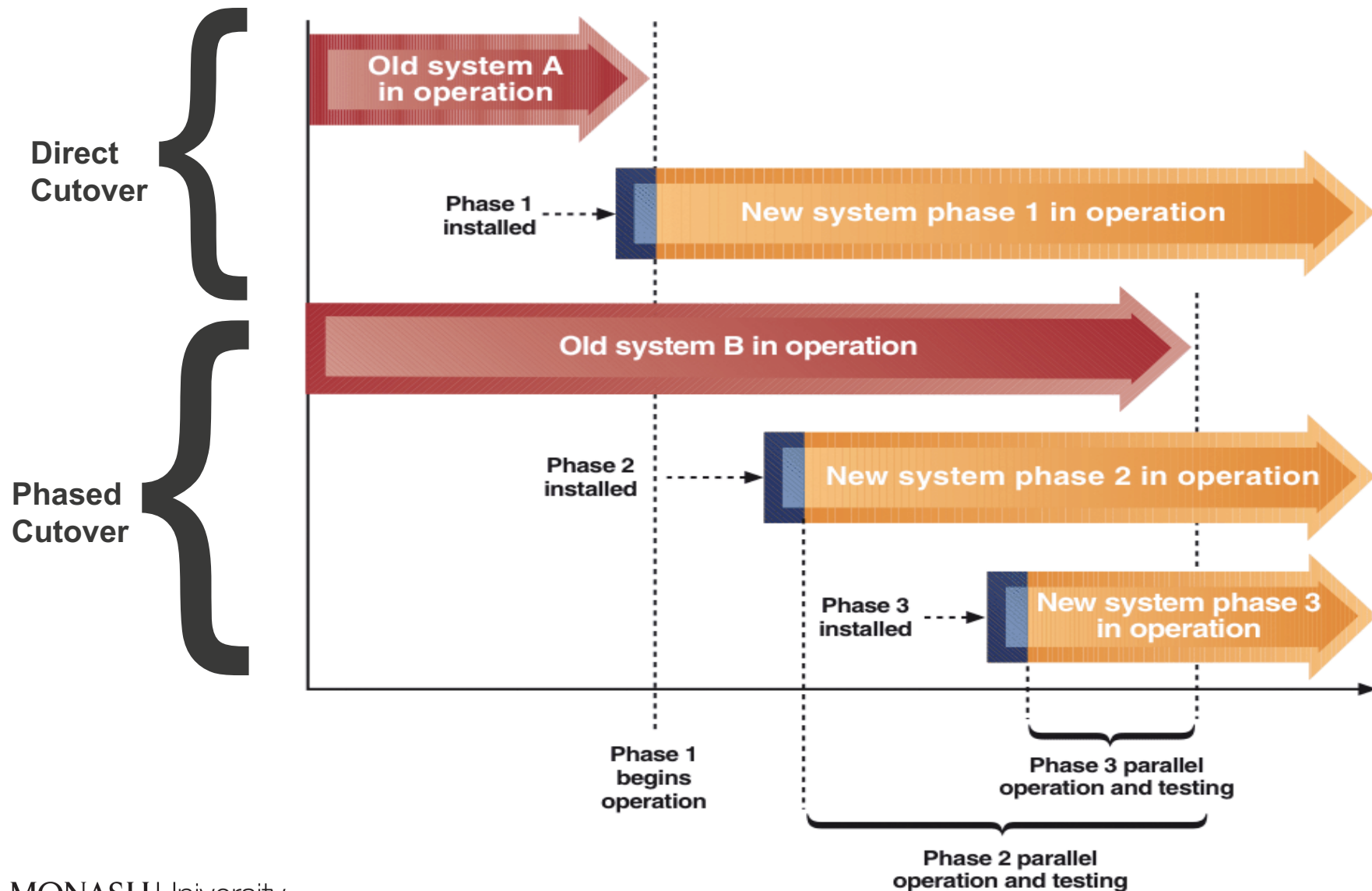
# Multiple locations - Pilot Installation



### Pilot installation

- Old and new systems operated concurrently
- Only part of the organisation tries out the new system
- The pilot system must prove itself at the test site
- Advantages
  - *Risks relatively low if problems occur*
  - *Errors are localised to pilot site*
  - *Can be used to train users before implementation at their own site*
- Disadvantages
  - *Lack of consistency between different parts of the organisation*

# Phased installation

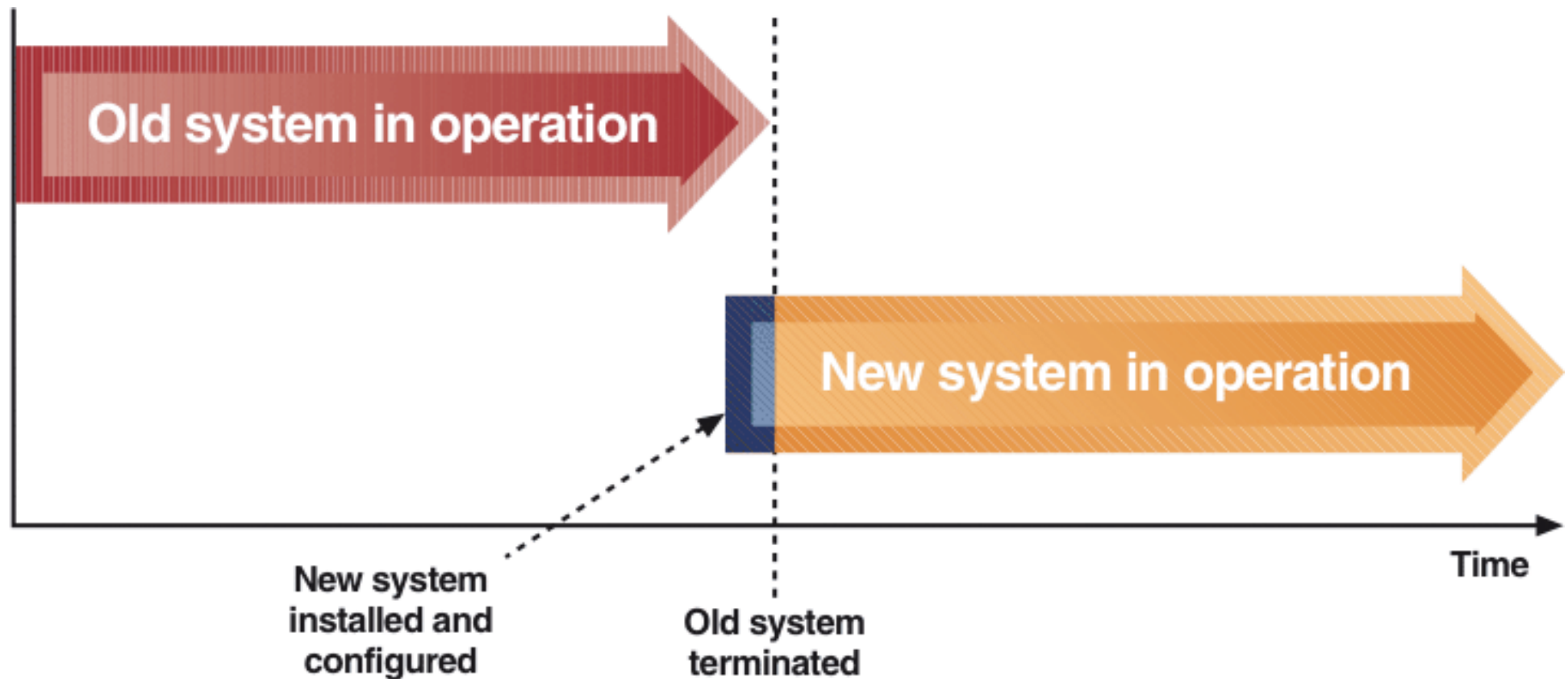




# Phased installation

- New system installed in series of steps or phase, where each phase adds components to the existing system
- Advantages
  - *Reduces risk because phase failure is less serious than system failure*
  - *Lower costs for earlier results*
  - *Benefits can be realised earlier*
  - *Rate of change minimised for users*
- Disadvantages
  - *Multiple phases cause more activities, milestones, and management complexity for entire effort*
  - *Close control of systems development is essential*
  - *Costs associated with the development of temporary interfaces to old systems*
  - *Limited business applicability*

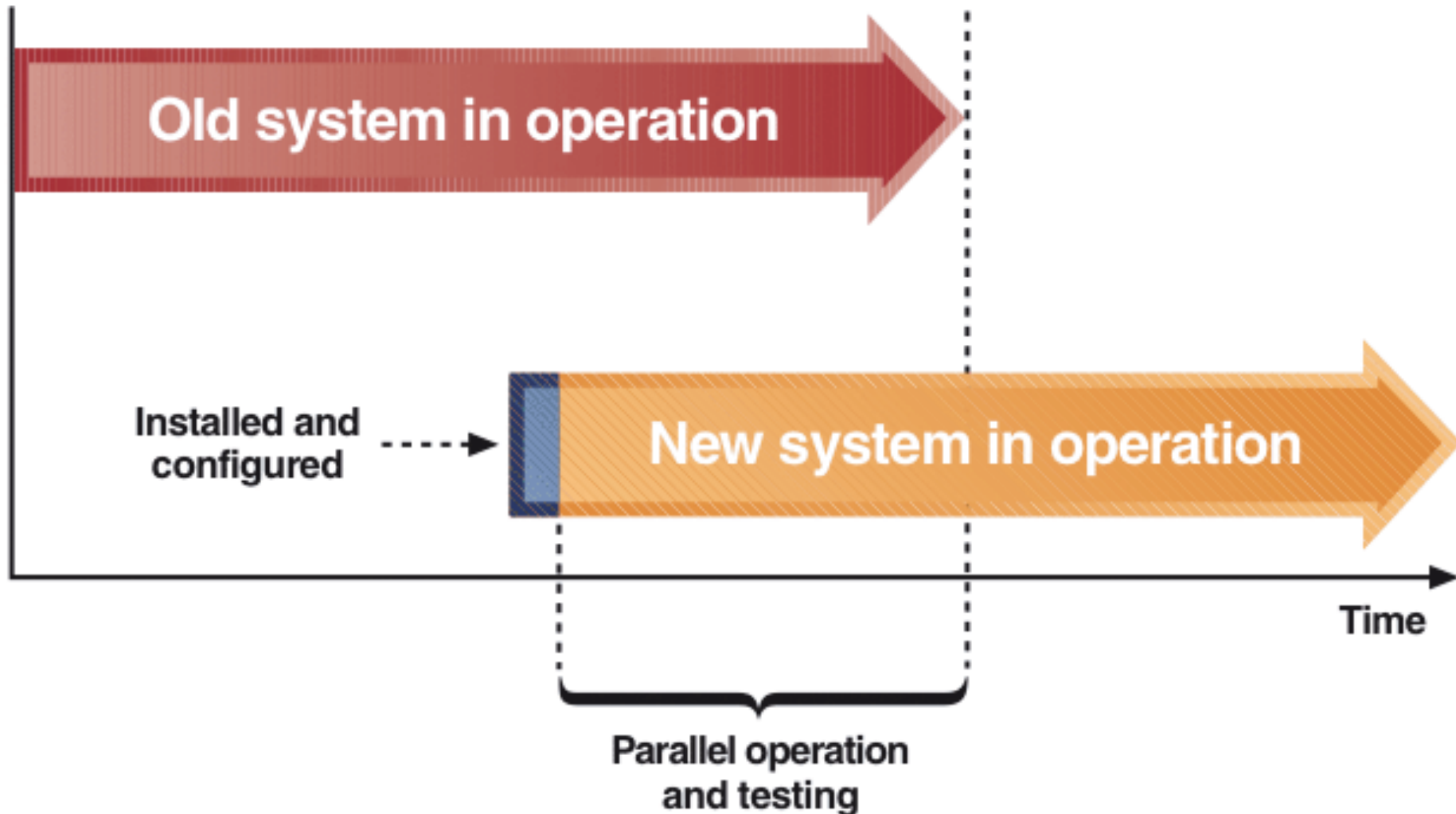
### Direct installation / Abrupt cutover



# Direct installation / Abrupt cutover

- This approach is meaningful when:
  - *the system is not replacing any other system*
  - *the old system is judged absolutely without value*
  - *the old system is either very small and/or very simple*
  - *the new system is completely different from the old and comparisons would be meaningless*
- Advantages
  - *Simple, fewer logistic issues to manage, costs minimised*
- Disadvantages
  - *High risk .. no backup*

### Parallel installation



# Parallel installation

- Old and new system both operate for an extended period of time. Cut over at the end of a business cycle, after balancing between both systems
- Advantages
  - *Risk low if problems occurs – continual backup*
- Disadvantages
  - *High cost: increased personnel, extra space, increased managerial and logistic complexity*

### Example:



Monash are going to update WES in February. Major changes. The University are experiencing financial issues due to reduced Government funding. Staff have been heavily involved with Usability testing and are looking forward to the rollout of the updated system. The system has to be rolled out across all local and international campuses. **What installation/deployment method would you recommend?**

**Consider:** Cost, System criticality, Disruption to business, User computer experience, System complexity, User resistance

# Monitor Operations

- Monitor user satisfaction
  - with functional requirements
  - with system performance
- Run benchmark tests
- Tune system

# Wrap up / Transition to Support

Support constitutes:

- Maintenance

1. *Corrective maintenance*
2. *Adaptive maintenance*
3. *Perfective maintenance*
4. *Preventative maintenance*

- Change management

- Post-implementation review

*.... after the system has been accepted*

- In 2010, maintenance up to 55% of the IT budget

*Forrester Research – 517 organisations*



# Corrective Maintenance

- Corrects analysis, design and implementation errors
  - *most corrective problems arise soon after installation or after major system changes*
  - *should have been isolated and corrected during development*
  - *professional practice during development should minimise the need (but will not remove it completely)*
  - *adds little or no value - focus on removing defects rather than adding anything new*
  - *accounts for up to 75% of all maintenance activity*

# Adaptive Maintenance

- To satisfy changes in the environment, changing business needs or new user requirements
  - *changes in tax laws, takeovers and mergers, new OS, etc*
  - *new type of report, new class of customer etc.*
- Less urgent - changes occur over time
- Adaptive maintenance is inevitable, does add value
- Maintenance staff need strong analysis and design skills as well as programming skills
  - *changes often require a complete SDLC*
  - *also need good understanding of the system*

# Perfective Maintenance

- To enhance performance, maintainability, usability
  - *adds desired features rather than required*
  - *better run times, faster transaction processing*
- To meet user requirements not previously recognised or given high priority
  - *missed in development or not known about*
  - *considered unimportant*
- Legacy systems (old systems running for at least 10 years) are likely candidates for perfective maintenance

# Preventative Maintenance

- Pay now or pay more later
  - defects or potential problems found and corrected before they cause any damage
  - reduce chance of future system failure
  - eg expand number of records beyond needs, standardise formats across platforms
- A natural by-product of maintenance work - identify and fix any potential problems noted while fixing other errors

# Change Management Systems

- Overall goal is to manage change effectively
- Organisations implement change management systems in an attempt to reduce the confusion and complexity of developing and maintaining systems

# Change Management Systems

- The aims of change management systems are
  - restrict access to production source and object code
  - reduce errors being introduced into production
  - single version of source and object code in production
  - improve quality and reliability of software
  - increase security and control
  - increase software productivity

# A change request example

Change Request			
Request Date	2/1/2007	Change Type	<input type="checkbox"/> Error Correction
Requested By	Wen-Hsu Chang, Comptroller		<input checked="" type="checkbox"/> Modification
Target System	Customer Accounts - Refunds		<input type="checkbox"/> New Function
<b>Change (or Error) Description</b>			
<p>U.S. check formats will soon change due to a recently enacted federal law. The new format reserves an area to the right of the current routing number to be used for a security bar code checksum.</p> <p>The law requires the new checksum to be printed on checks dated on or after January 1, 2008.</p> <p>We currently use a portion of the area in question to print a multicolored security symbol. The security symbol will need to be moved or eliminated, and the security bar code checksum will have to be added.</p>			

# A change review form

Change Review			
Change Request ID	2007-11	Date Reviewed	2/7/2007
Priority	<input type="checkbox"/> Critical <input checked="" type="checkbox"/> Necessary <input type="checkbox"/> Optional		
<b>Hardware Implications</b> need to verify ability of current printers to write a security bar code in mandated area			
<b>Software Implications</b> database will need to be modified to store the security bar code with other check information check writing program must be modified to generate and print the security bar code			
<b>Performance Implications</b> none			
<b>Operating Budget Implications</b> none			
<b>Other Implications</b> none			
Disposition	<input checked="" type="checkbox"/> Approved <input type="checkbox"/> Rejected <input type="checkbox"/> Suspended		
Reason			
Latest Implementation Date		12/31/2007	
Reevaluation Date	n/a	Signature	



# System closure

## Successful systems closure involves:

- *Gaining stakeholder and customer acceptance of the system - ensure that outcomes are consistent with those defined in the Acceptance criteria (including negotiated variances)*
- *Ensuring all legally binding agreements are met, payments and collections made*
- *Acknowledgement of the success of the team, and dealing with an emotional / career issues team members may face, finalise performance management reports*
- *Internal review of the system to capture lessons learnt*
- *Final Report / Presentation to key stakeholders to formally announce the end of the system development*

# Post-Implementation Review

- A PIR analyses what went right and wrong with a project. It is conducted 2 to 6 months after conversion by a team which includes user reps, development staff, internal auditors and sometimes external consultants - development team is not in charge!
  - *look at original requirements and objectives*
  - *evaluate how well they were met*
  - *compare costs of development and operation against original estimates (maintenance costs ??)*
  - *compare original and actual benefits*
  - *new system reviewed to see whether more of original or additional benefits can be realised*

Must not  
become a  
witch-hunt



What went wrong ???  
Learn for the future !!!



## Workshop Preparation

The Workshop for Week 12 will include:

Assignment 3 Presentations

Exam Revision

Moodle Quiz for Week 10 & 11 Seminars

**Thanks for watching**

# Resources:

Satzinger, J. W., Jackson, R.B., Burd, S.D. and R. Johnson (2016)  
Systems Analysis and Design in a Changing World, 7<sup>th</sup> Edition,  
Thomson Course Technology, Chapter 14, pp. 454-473

Deloitte (2013): The Outsourcing Handbook – A guide to  
outsourcing