Murdoch
UNIVERSITY

From analysis to design

Topic 6

ICT284 Systems Analysis and Design

About this topic

The previous topics have covered systems analysis, where the purpose was to understand and specify *what* the new system should accomplish. There are choices involved in *how* the system will be implemented based on the requirements, and these are evaluated, determined and documented in the design stage.

Design thus provides a bridge between analysis and implementation: just as analysis provides the starting point for design, design provides the starting point for implementation. Like analysis, design is a model building activity, but whereas analysis built models to capture requirements, design builds models to represent the solution.

Unit learning outcomes addressed in this topic

1. Explain how information systems are used within organisations to fulfil organisational needs
2. **Describe the phases and activities typically involved in the systems development life cycle**
3. Describe the professional roles, skills and ethical issues involved in systems analysis and design work
4. Use a variety of techniques for analysing and defining business problems and opportunities and determining system requirements
5. Model system requirements using UML, including use case diagrams and descriptions, activity diagrams and domain model class diagrams
6. **Explain the activities involved in systems design, including designing the system environment, application components, user interfaces, database and software**
7. Represent early system design using UML, including sequence diagrams, architectural diagrams and design class diagrams
8. Describe tools and techniques for planning, managing and evaluating systems development projects
9. Describe the key features of several different systems development methodologies
10. **Present systems analysis and design documentation in an appropriate, consistent and professional manner**

Murdoch UNIVERSITY

Topic learning outcomes

After completing this topic you should be able to:

- Describe systems design and contrast it with systems analysis
- List the documents and models used as inputs to or output from systems design
- Describe each major design activity
- Describe how different alternative solutions may be evaluated for feasibility
- Discuss the advantages and disadvantages involved in the ‘buy versus build’ decision for software components or applications
- Explain why security and integrity methods and controls need to be considered throughout all design activities
- Briefly describe some security and integrity methods and controls

Murdoch UNIVERSITY

Resources for this topic

READING

- Satzinger, Jackson & Burd, Chapter 6

  (read the section 'System Controls and Security' to get the main concepts, but you don't need to remember all of all the detail)

- Satzinger, Jackson & Burd, Chapter 11 section 'Determine Project Risk and Feasibility' p.343-345

Except where otherwise referenced, all images in these slides are from those provided with the textbook: Satzinger, J., Jackson, R. and Burd, S. (2016) *Systems Analysis and Design in a Changing World*, 7th edition, Course Technology, Cengage Learning: Boston. ISBN-13 9781305117204

Tutorial 6 –
Extending the use case models

In this tutorial, we'll complete our coverage of requirements modelling by creating fully developed use case descriptions along with activity diagrams and system sequence diagrams to describe use cases in more depth. These models will form a basis for the design models that we will go on to create.

Topic outline

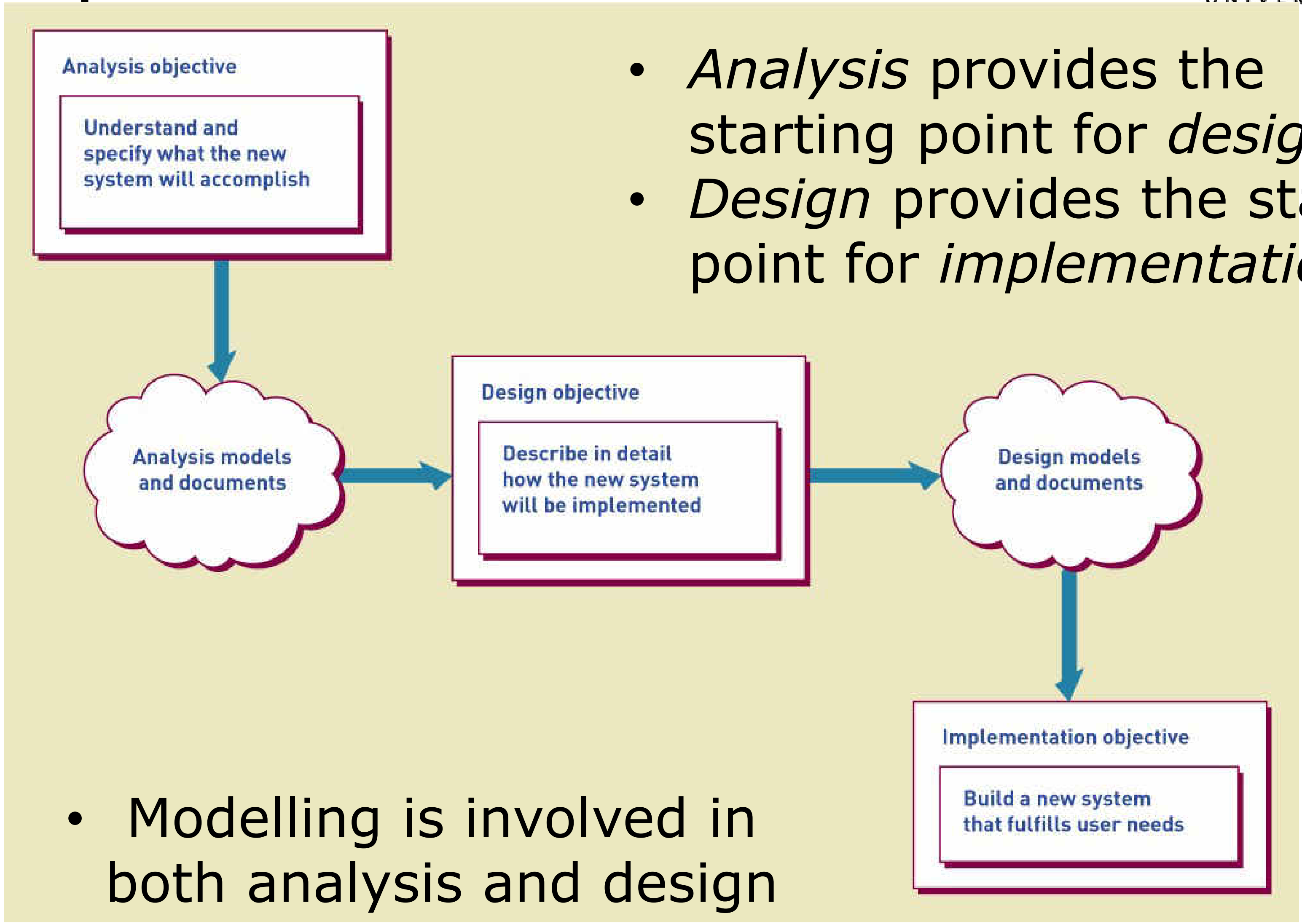
- Introduction to systems design
- Overview of systems design activities
- Evaluating alternative solutions
- Overview of system controls and security

Introduction to systems design

Overview

- Analysis says “*what*” is required and design tells us “*how*” the system will be configured and constructed
- Topics 2, 3, 4 and 5 covered *systems analysis* activities (discover and understand details)
- This topic introduces *systems design* and the design activities involved in systems development
- We’ll continue to cover the design activities in more detail in topics 7, 8 and 9

Analysis to Design to Implementation


Analysis objective
Understand and specify what the new system will accomplish
Analysis models and documents
Design objective
Describe in detail how the new system will be implemented
Design models and documents
Implementation objective
Build a new system that fulfills user needs

- *Analysis* provides the starting point for *design*
- *Design* provides the starting point for *implementation*

- Modelling is involved in both analysis and design

Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

What is systems design?

- Analysis provides the starting point for design
- Design provides the starting point for implementation
- Whereas systems analysis places emphasis on the business problem…
- … Systems design places emphasis on the technical or implementation concerns of the system
- Objective of design is *to define, organize, and structure the components of the final solution to serve as a blueprint for construction*

Choices in systems design

- System and software architecture - high level structure of components and how the software is distributed across the hardware
- Meeting the non-functional requirements – how to implement the requirements for usability, reliability, performance, security, etc?
- Buy versus build
  - whole system v components

Modelling in systems design

- Like analysis, design is a model building activity
- Whereas analysis built models to capture requirements, design builds models to represent the solution
- The formality of the project will dictate the type, complexity, and depth of models
- Agile/iteration projects typically build fewer models, but models are still created
- Jumping to programming without design often causes less than optimum solutions and may require rework

Modelling in systems design

- Converts functional models from analysis into models that represent the solution
  - Focus is on technical issues
  - Requires less user involvement than analysis

- Design may use structured or OO approaches
  - e.g. Database can be relational, OO, or hybrid
  - We'll continue to look at OO approaches in this unit

Analysis to design models

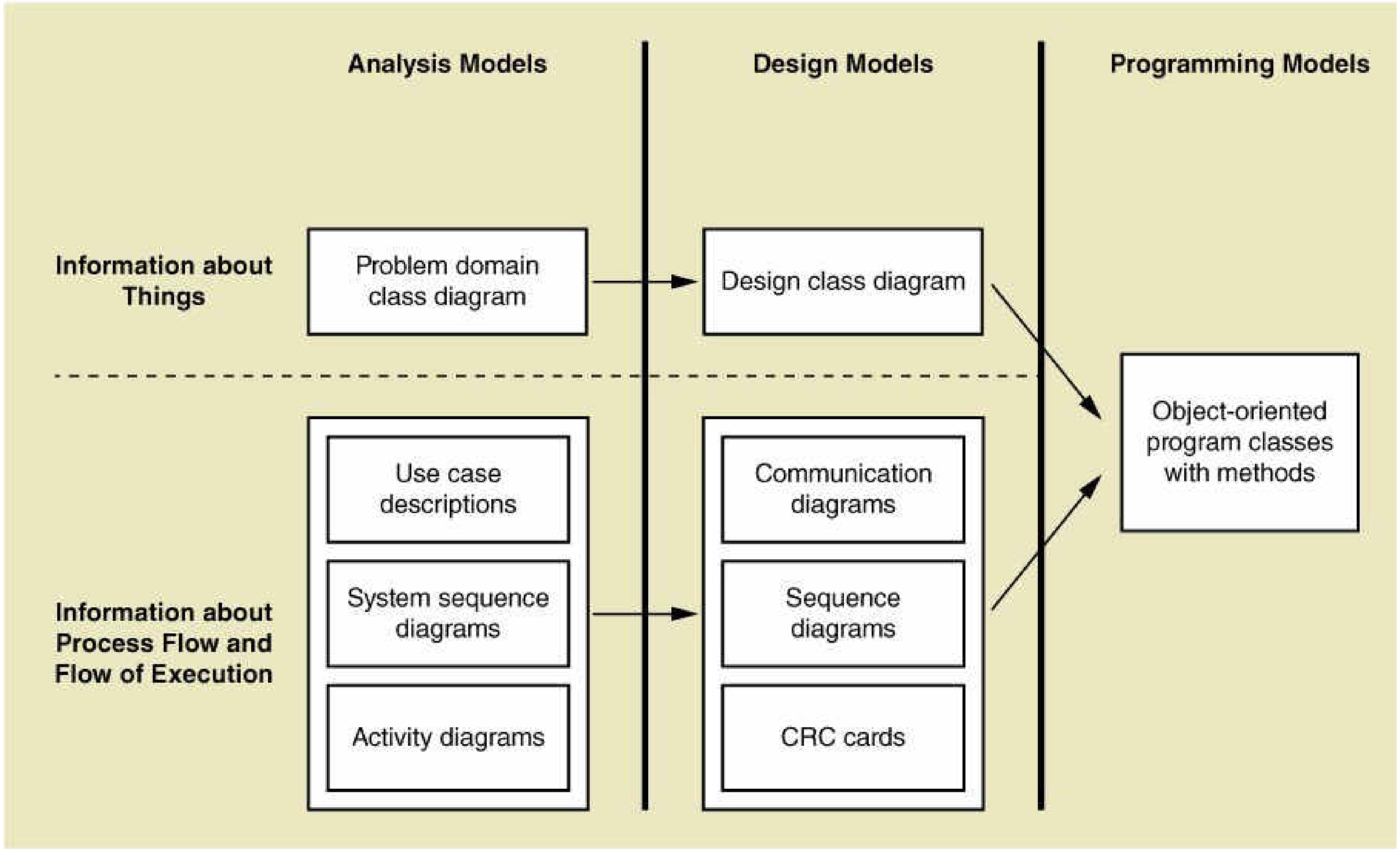


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Analysis to design models

- We continue to extend and develop the models from the requirements stage into the design stage
- All of the models must be consistent and integrate together to provide a complete picture

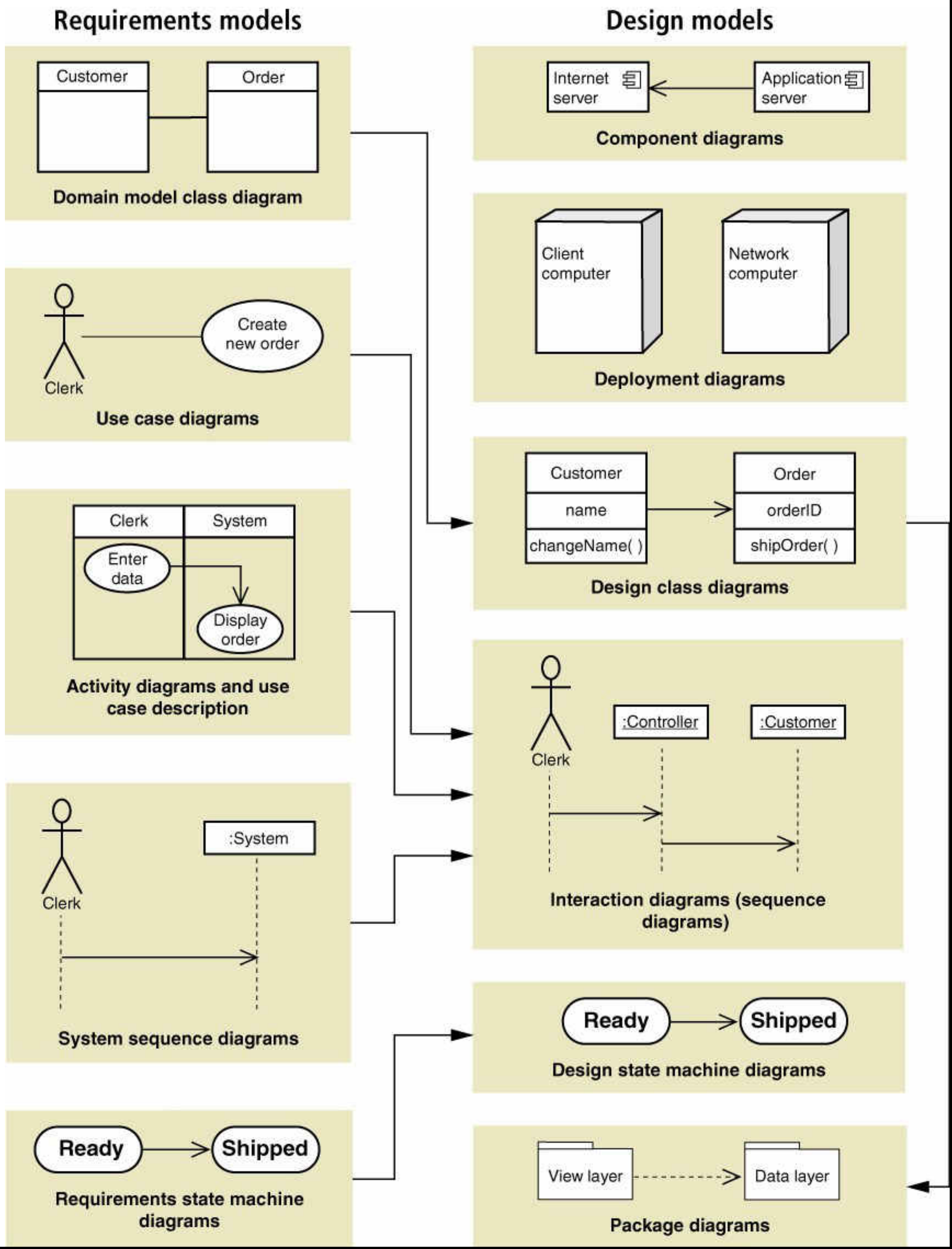


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Summing up…

- Analysis says *what* is required and design tells us *how* the system will be configured and constructed
- Like analysis, design is a model building activity – but whereas analysis built models to capture *requirements*, design builds models to represent the *solution*
- We continue to extend and develop the models from the requirements stage into the design stage

Design activities

Design activities

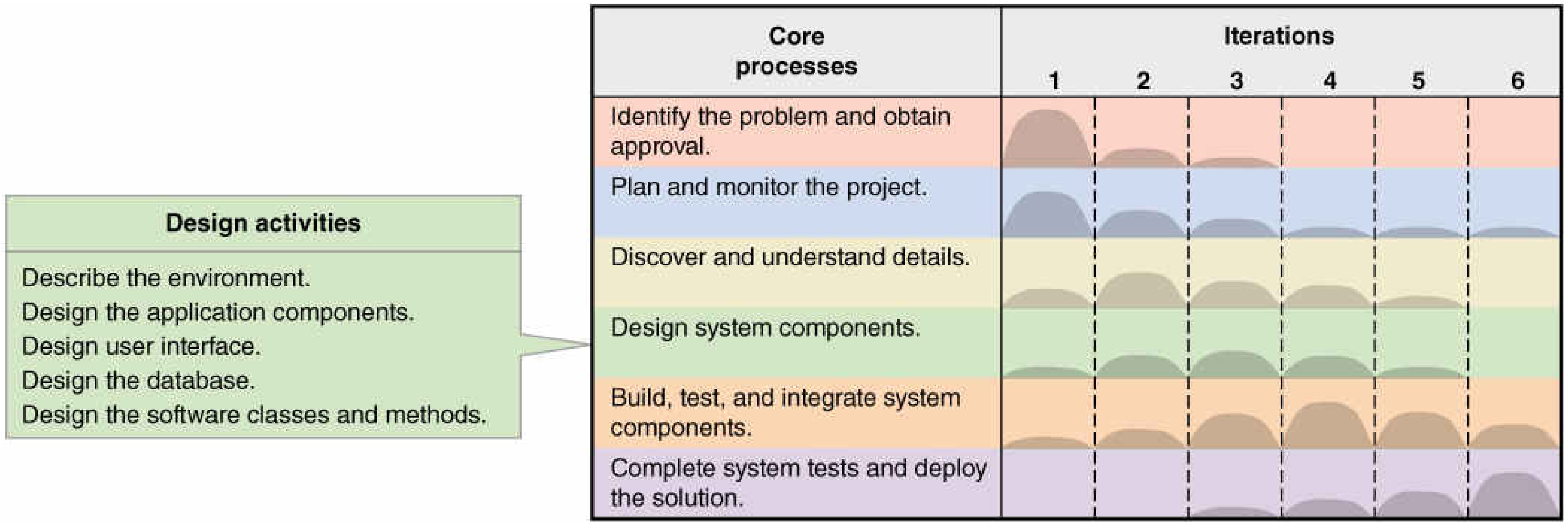
There are five main design activities, corresponding to components of the new system:

- The environment
- Application components
- User interface
- Database
- Software classes and methods

These tend to be done in parallel, as decisions made in one activity will influence the others

And system controls and security must be considered throughout ALL design activities

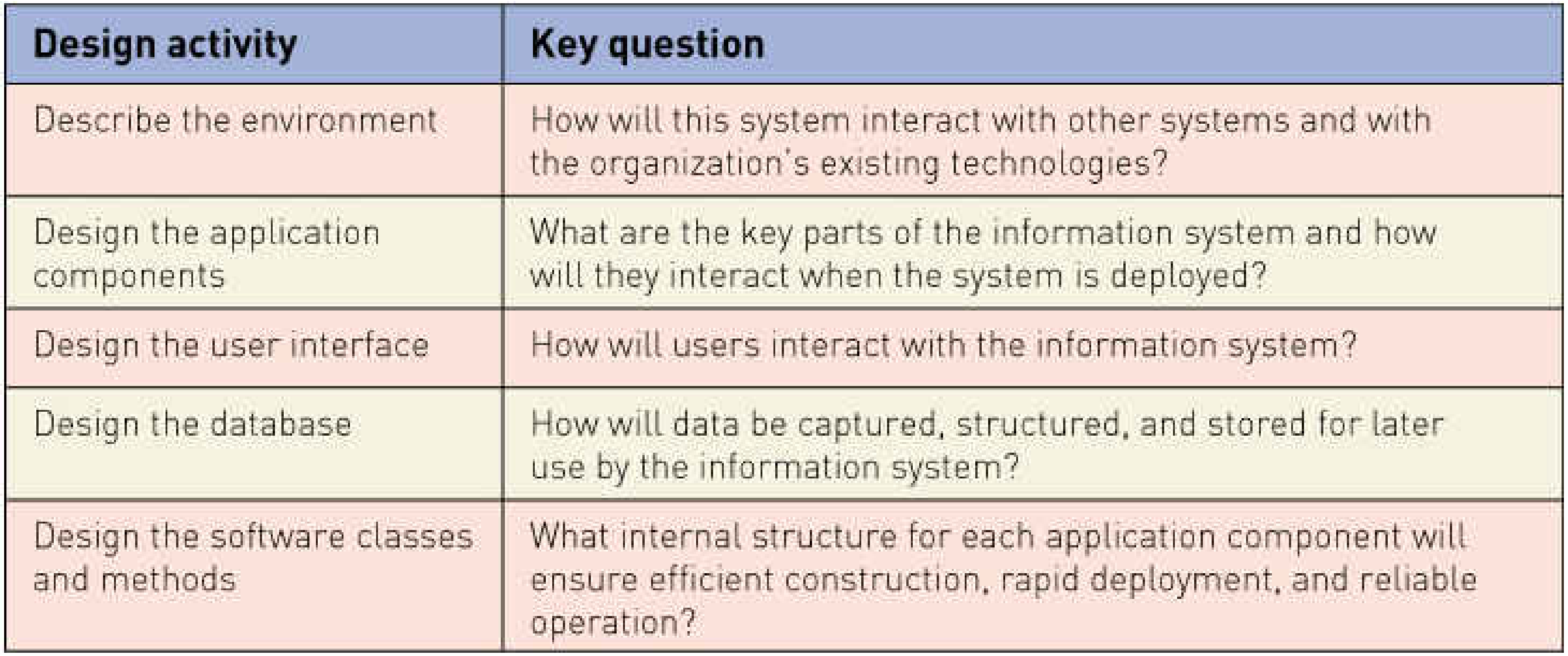
Design activities and iterations



Design activities
Describe the environment.
Design the application components.
Design user interface.
Design the database.
Design the software classes and methods.
Core processes
Iterations
1
2
3
4
5
6
Identify the problem and obtain approval.
Plan and monitor the project.
Discover and understand details.
Design system components.
Build, test, and integrate system components.
Complete system tests and deploy the solution.

Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Key design questions for each activity



Design activity	Key question
Describe the environment	How will this system interact with other systems and with the organization's existing technologies?
Design the application components	What are the key parts of the information system and how will they interact when the system is deployed?
Design the user interface	How will users interact with the information system?
Design the database	How will data be captured, structured, and stored for later use by the information system?
Design the software classes and methods	What internal structure for each application component will ensure efficient construction, rapid deployment, and reliable operation?

Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Describe the environment

Murdoch UNIVERSITY

How will this system interact with other systems and with the organisation's existing technologies?

- A system will rarely be completely new – it will have to fit with what already exists in the organisation, so this can limit design choices
- Two key elements of environment:
  - External systems
  - Technology architecture

Describe the environment

1. Need information about communication flows to and from *external systems* this system must interface with:
   - Message formats
   - Web and network addresses of sources/destinations
   - Communication protocols
   - Security methods
   - Error detection and recovery
2. Solution needs to fit with the organisation's existing *technology architecture* – all its hardware, software, networks etc
   - Discover and describe existing architecture

More in Topic 7

Design the application components

What are the key parts of the information system and how will they interact when the system is deployed?

- Application component is a well defined unit of software that performs some function(s)
- Need to decide how the functionality is to be packaged into components, and how they will interact, e.g.
  - Subsystems
  - Database
  - Separation into layers

More in Topic 7

- Application component design is precursor to defining the software classes and methods

Design the application components

- Some issues involved in designing the application components:
  - Scope and size – what are the functions, boundaries, interfaces?
  - Programming language – what are the accepted languages?
  - Build or buy – is an acceptable version available to purchase?

More in Topic 7

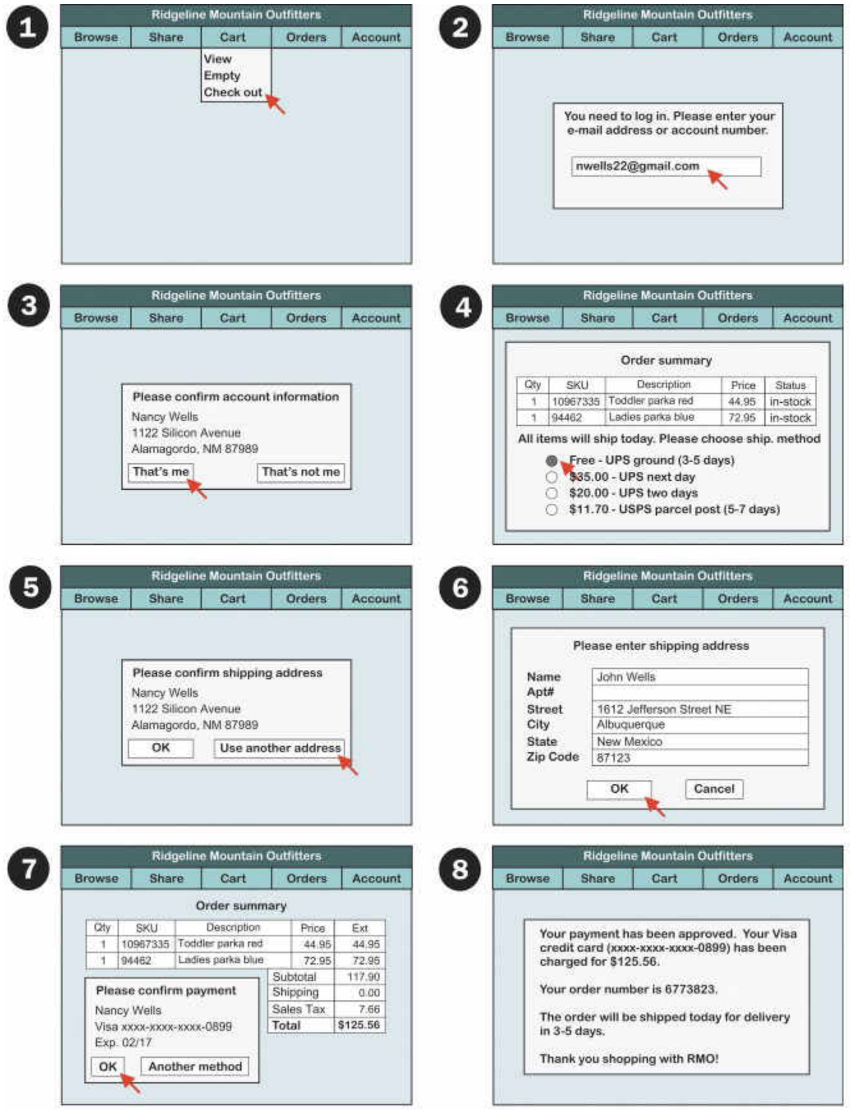
Design the user interface

How will users interact with the information system?

- The user interface has a large impact on user productivity – to the user, the interface IS the system
- User involvement throughout crucial to success
- Designing the interface requires analysis techniques to determine user needs, as well as design activities focused on devices and software
- Current needs require multiple user interfaces for many different devices and environments

More in Topic 8

Typical models for user interface design – storyboard, system sequence diagram, screen prototypes



1

Ridgeline Mountain Outfitters

Browse | Share | Cart | Orders | Account

View
Empty
Check out

2

Ridgeline Mountain Outfitters

Browse | Share | Cart | Orders | Account

You need to log in. Please enter your e-mail address or account number.

nwells22@gmail.com

3

Ridgeline Mountain Outfitters

Browse | Share | Cart | Orders | Account

Please confirm account information

Nancy Wells
1122 Silicon Avenue
Alamagordo, NM 87989

That's me | That's not me

4

Ridgeline Mountain Outfitters

Browse | Share | Cart | Orders | Account

Order summary

Qty	SKU	Description	Price	Status
1	10967335	Toddler parka red	44.95	in-stock
1	94462	Ladies parka blue	72.95	in-stock

All items will ship today. Please choose ship. method

- ● Free - UPS ground (3-5 days)
- ○ $35.00 - UPS next day
- ○ $20.00 - UPS two days
- ○ $11.70 - USPS parcel post (5-7 days)

5

Ridgeline Mountain Outfitters

Browse | Share | Cart | Orders | Account

Please confirm shipping address

Nancy Wells
1122 Silicon Avenue
Alamagordo, NM 87989

OK | Use another address

6

Ridgeline Mountain Outfitters

Browse | Share | Cart | Orders | Account

Please enter shipping address

Name	John Wells
Apt#	
Street	1612 Jefferson Street NE
City	Albuquerque
State	New Mexico
Zip Code	87123

OK | Cancel

7

Ridgeline Mountain Outfitters

Browse | Share | Cart | Orders | Account

Order summary

Qty	SKU	Description	Price	Ext
1	10967335	Toddler parka red	44.95	44.95
1	94462	Ladies parka blue	72.95	72.95
			Subtotal	117.90
			Shipping	0.00
			Sales Tax	7.66
			Total	$125.56

Please confirm payment

Nancy Wells
Visa xxxx-xxxx-xxxx-0899
Exp. 02/17

OK | Another method

8

Ridgeline Mountain Outfitters

Browse | Share | Cart | Orders | Account

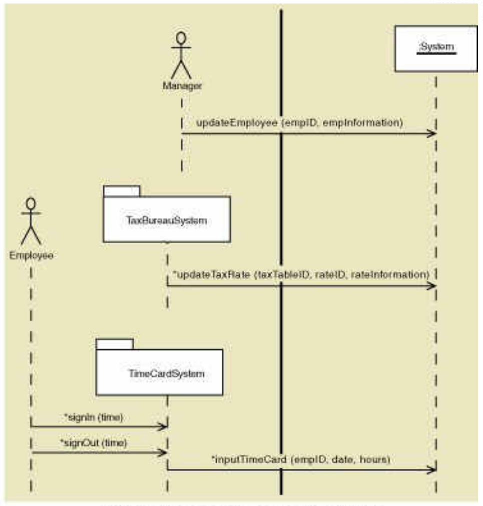
Your payment has been approved. Your Visa credit card (xxxx-xxxx-xxxx-0899) has been charged for $125.56.

Your order number is 6773823.

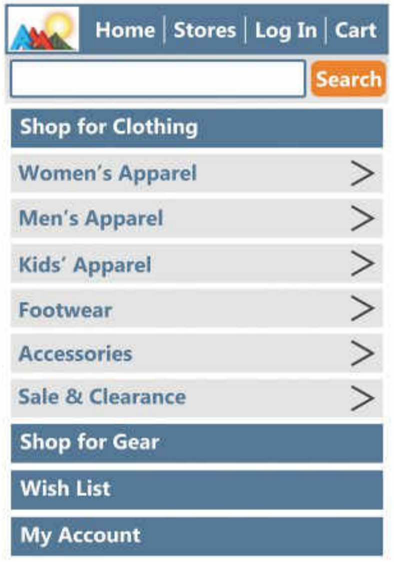
The order will be shipped today for delivery in 3-5 days.

Thank you shopping with RMO!

Storyboard


:System
Manager
updateEmployee (empID, empInformation)
TaxBureauSystem
Employee
*updateTaxRate (taxTableID, rateID, rateInformation)
TimeCardSystem
*signIn (time)
*signOut (time)
*inputTimeCard (empID, date, hours)

System sequence diagram



Small screen menu prototype

Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Design the database

How will data be captured, structured an stored for later use by the information system?

- By definition, an information system requires a data repository – usually in a database
- Current technology frequently uses Relational Database Management Systems (RDBMS)
- Requires converting the data model (from the DMCD or ERD) to a relational database
- Database design also involves addressing many other technical issues, such as performance and security
- A bit more in Topic 7, and a lot more in ICT285

Design software classes and methods

What internal structure for each application component will ensue efficient construction, rapid deployment, and reliable operation?

- Sometimes called Detailed design
- Define and design the classes and methods needed to carry out each use case
- A model building activity:
  - Design Class Diagrams
  - Sequence Diagrams
  - State Machine Diagrams

More in Topic 10

Typical Design Class Diagram with attributes and methods

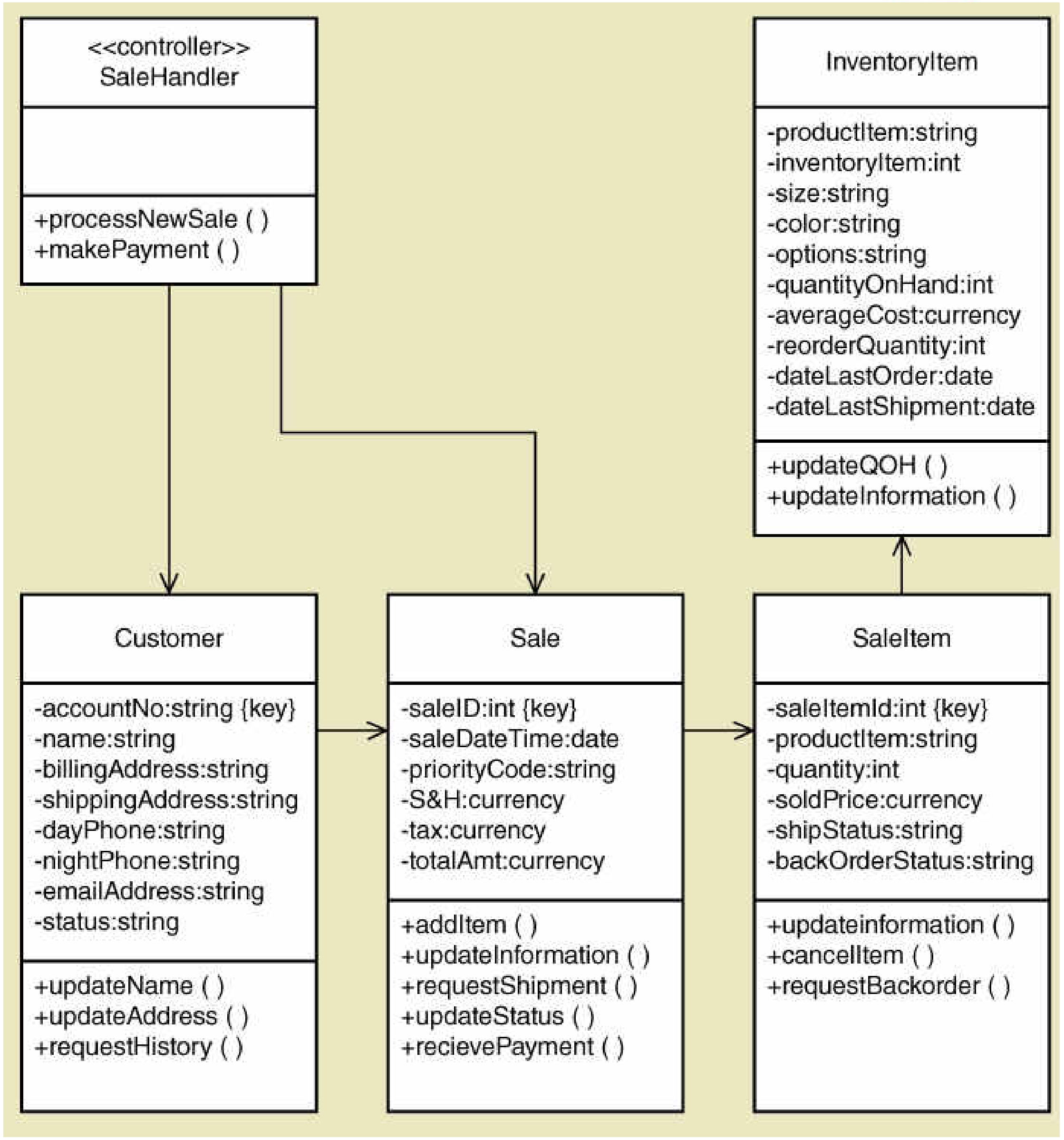


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Summing up…

- There are five main design activities, corresponding to components of the new system:
    - The environment
    - Application components
    - User interface
    - Database
    - Software classes and methods
- These tend to be done in parallel, as decisions made in one activity will influence the others
- System controls and security must be considered throughout ALL design activities

Murdoch
UNIVERSITY

Feasibility and evaluating alternatives

Determining feasibility

Feasibility is the measure of how beneficial or practical the development of an information system will be to an organisation

Feasibility analysis is the process by which feasibility is measured

The **creeping commitment** approach to feasibility proposes that feasibility should be measured and re-evaluated throughout the system development life cycle

A project can be cancelled or revised at any check point, despite what resources have been spent

Feasibility assessment and choosing alternatives

In project planning:

- Very preliminary estimate of feasibility

In analysis and design:

- Alternative solutions are defined in terms of scope and level of automation
- Each alternative solution is analysed for feasibility
- Further evaluation of implementation alternatives is made

Feasibility criteria

Main feasibility criteria:

- Organisational feasibility
- Technical feasibility
- Resource & economic feasibility
- Schedule feasibility

Also:

- Legal feasibility

Organisational feasibility

Operational, **organisational** and cultural **feasibility** focuses on whether the system will work in a given organisational climate

- Does management support the system?
- How do end users feel about their role in the system?
- What end users may resist or not use the system? How can this be overcome?
- How will the working environment change?
- Can users and management adapt to the change?

Technical feasibility

Technical feasibility is a measure of the practicality of a specific technical solution and the availability of technical resources and expertise.

- Is the proposed technology or solution practical?
- Do we currently possess the necessary technology?
- Do we have the necessary technical expertise?

Resource & economic feasibility

After identifying the **resources** required, economic **feasibility** is a measure of the cost-effectiveness of a project or solution,

- Are the resources (human and physical) available?
- Will the benefits of the system outweigh the costs?
  - Put a dollar value to all the costs and benefits associated with the system
  - Use these figures to perform some type of **cost/benefit analysis**

Schedule feasibility

Murdoch
UNIVERSITY

Schedule feasibility is a measure of how reasonable the project timetable is

- Can the solution be designed and implemented within an acceptable time period?
- Need to determine whether deadlines are mandatory or desirable, and what impact they have on the work scoped

Legal feasibility

Legal feasibility is a measure of how well a potential solution can be implemented within existing legal and contractual obligations:

- Copyrights
- Legal requirements for financial reporting
- Trade practice laws
- National data and work laws

Alternative solutions

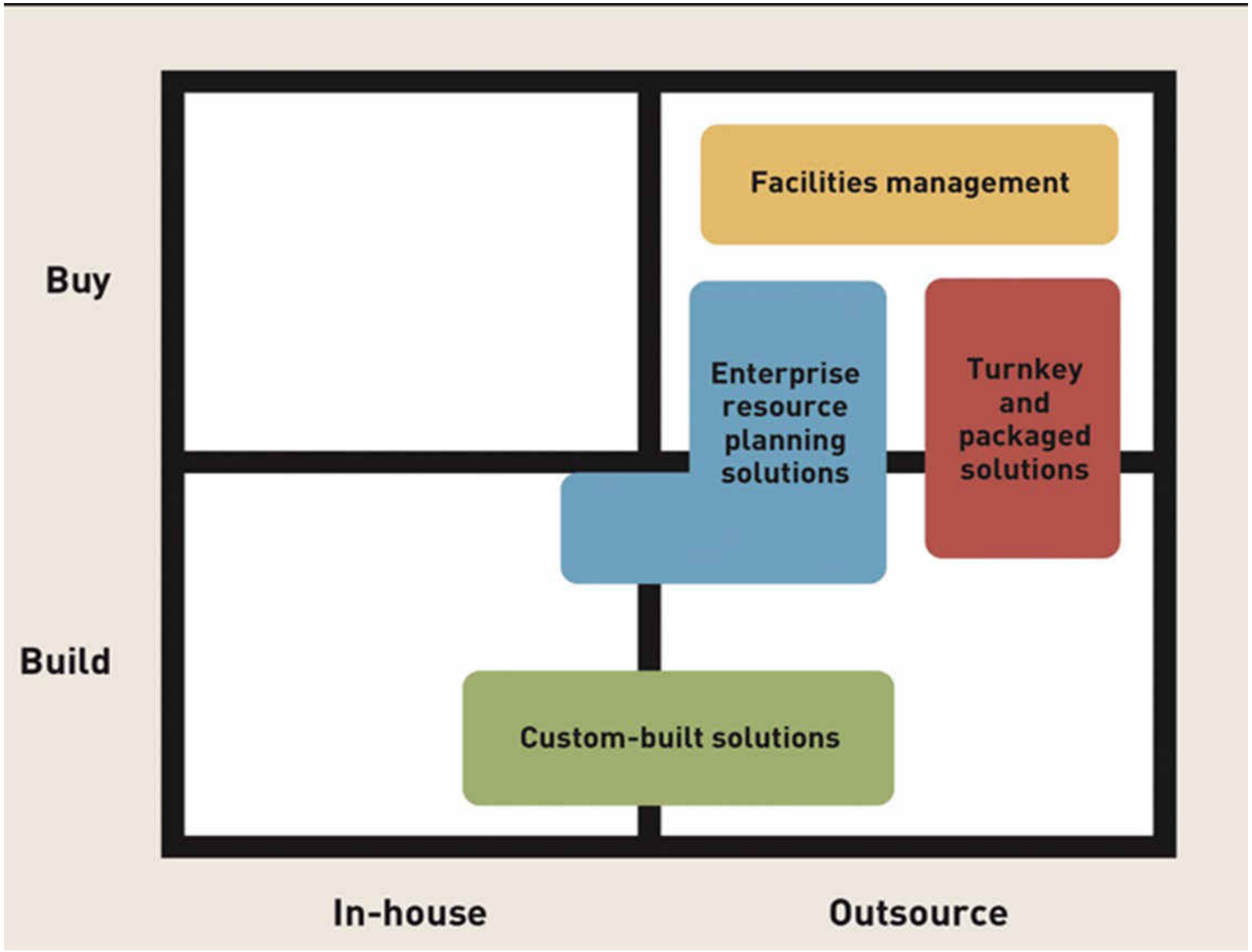
Variations on obtaining system

- Facilities management or Service Provider Solutions – *outsource all IS support*
- Packaged software, turnkey system, enterprise resource planning (ERP) system
- Custom-built software systems
- In-house development
  - Selection dimensions

    Buy vs. build

    In-house vs. outsource

Types of alternative solutions



Buy
Build
In-house
Outsource
Facilities management
Enterprise resource planning solutions
Turnkey and packaged solutions
Custom-built solutions

Image from Satzinger, J. Jackson, R. & Burd, S. (2004) Systems Analysis and Design in a Changing World, 3rd edition. Course Technology, Thomson Learning. Figure 8.6

Build vs buy decision: BUILD

FOR

- Gives you exactly what you want
- Helps you innovate where others don’t
- Allows easy incorporation of 3rd party solutions
- Adapts and pivots with you
- Improves valuation/competitive advantage
- Secure

This list copied from https://www.fivetalent.com/build-vs-buy-weighing-custom-software-against-off-the-shelf-solutions/

Build vs buy decision: BUILD

AGAINST

- Higher cost
- Potentially longer implementation
- Need to understand your processes first
- Requires staff or vendor support
- Once you’re in, hard to let go of

This list copied from https://www.fivetalent.com/build-vs-buy-weighing-custom-software-against-off-the-shelf-solutions/

Build vs buy decision: BUY

FOR

- Lower up front costs
- Defines processes for you
- Follows industry practices, which creates a workforce skilled in the packaged solution
- Time to market could be faster
- Overall market pressures can drive useful new features
- Customer support

This list copied from https://www.fivetalent.com/build-vs-buy-weighing-custom-software-against-off-the-shelf-solutions/

Build vs buy decision: BUY

AGAINST

- Rigid
- End up with external manual processes and silos of data if it doesn’t fit your processes
- Target for hackers
- Costly to customize
- Exporting data is messy and expensive
- Lowest common denominator for feature sets means there are numerous features you’ll never use

This list copied from https://www.fivetalent.com/build-vs-buy-weighing-custom-software-against-off-the-shelf-solutions/

Putting it all together

A **feasibility analysis matrix (evaluation matrix)** can provide a comparison and ranking of the candidate systems

	Weighting	Candidate 1	Candidate 2	Candidate 3
Description				
Organisational Feasibility				
Technological Feasibility				
Resource Feasibility				
Schedule Feasibility				
Legal Feasibility				
Ranking				

Adapted from from Whitten, J.L., Bentley, L.D. & Dittman, K.C., (2004), Systems Analysis and Design Methods, 6th Edition, McGraw-Hill. Figure 10.9

Sample feasibility analysis matrix

Feasibility Criteria	Wt.	Candidate 1	Candidate 2	Candidate 3
Organisational Feasibility A description of to what degree the candidate would benefit the organisation and how well the system would work in it. Includes how well received this solution would be from both user management, user, and organisation perspective.	**30%**	Only supports Member Services requirements Current business processes would have to be modified to take advantage of software functionality. **Score: 60**	Fully supports user required functionality. **Score: 100**	Same as candidate 2. **Score: 100**
Technological Feasibility **Technology.** An assessment of the maturity, availability (or ability to acquire), and desirability of the computer technology needed to support this candidate. **Expertise.** An assessment of the technical expertise needed to develop, operate, and maintain the candidate system.	**30%**	Current production release of Platinum Plus package is version 1.0 and has only been on the market for 6 weeks. Maturity of product is a risk and company charges an additional monthly fee for technical support. Required to hire or train C++ expertise to perform modifications for integration requirements. **Score: 50**	Although current technical staff has only Powerbuilder experience, the senior analysts who saw the MS Visual Basic demonstration and presentation have agreed the transition will be simple and finding experienced VB programmers will be easier than finding Powerbuilder programmers and at a much cheaper cost. MS Visual Basic is a mature technology based on version number. **Score: 95**	Although current technical staff is comfortable with Powerbuilder, management is concerned with recent acquisition of Powerbuilder by Sybase Inc. MS SQL Server is a current company standard and competes with SYBASE in the client/server DBMS market. Because of this we have no guarantee future versions of Powerbuilder will "play well" with out current SQL Server. **Score: 60**
Resource/Economic Feasibility **Cost to develop:** **Payback period (discounted):** **Net present value:** **Detailed calculations:**	**30%**	Approximately $350,000. Approximately 4.5 years. Approximately $210,000. See Attachment A. **Score: 60**	Approximately $418,040. Approximately 3.5 years. Approximately $306,748. See Attachment A. **Score: 85**	Approximately $400.000. Approximately 3.3 years. Approximately $325,500. See Attachment A. **Score: 90**
Schedule Feasibility An assessment of how long the solution will take to design and implement.	**10%**	Less than 3 months. **Score: 95**	9-12 months. **Score: 80**	9-12 months. **Score: 85**
Ranking	**100%**	**60.5**	**92**	**85**

Murdoch
UNIVERSITY

Alternative solutions - researching options for ‘Buy’ solutions

Identify specifications that are important to the hardware and/or software to be purchased

Identify potential vendors and products and obtain the necessary information about these products

- *Internal standards* may exist for hardware and software selection.
- *Information services* are primarily intended to constantly survey the marketplace for new products and advise prospective buyers on what specifications to consider.
- *Trade newspapers and periodicals* offer articles and experiences on various types of hardware and software that you may be considering.

Can issue a **request for information** (RFI)

Alternative solutions - solicit proposals from vendors

If the organisation isn't already committed to a single source, get proposals or quotes from vendors:

An **RFQ** (request for quotation) solicits quotations for specific products from different vendors

An **RFP** (request for proposal) communicates requirements and desired features to vendors, who then respond with proposals

Alternative solutions - evaluate and rank vendor proposals

- Establish evaluation criteria and scoring system *before* actual evaluation takes place to reduce bias

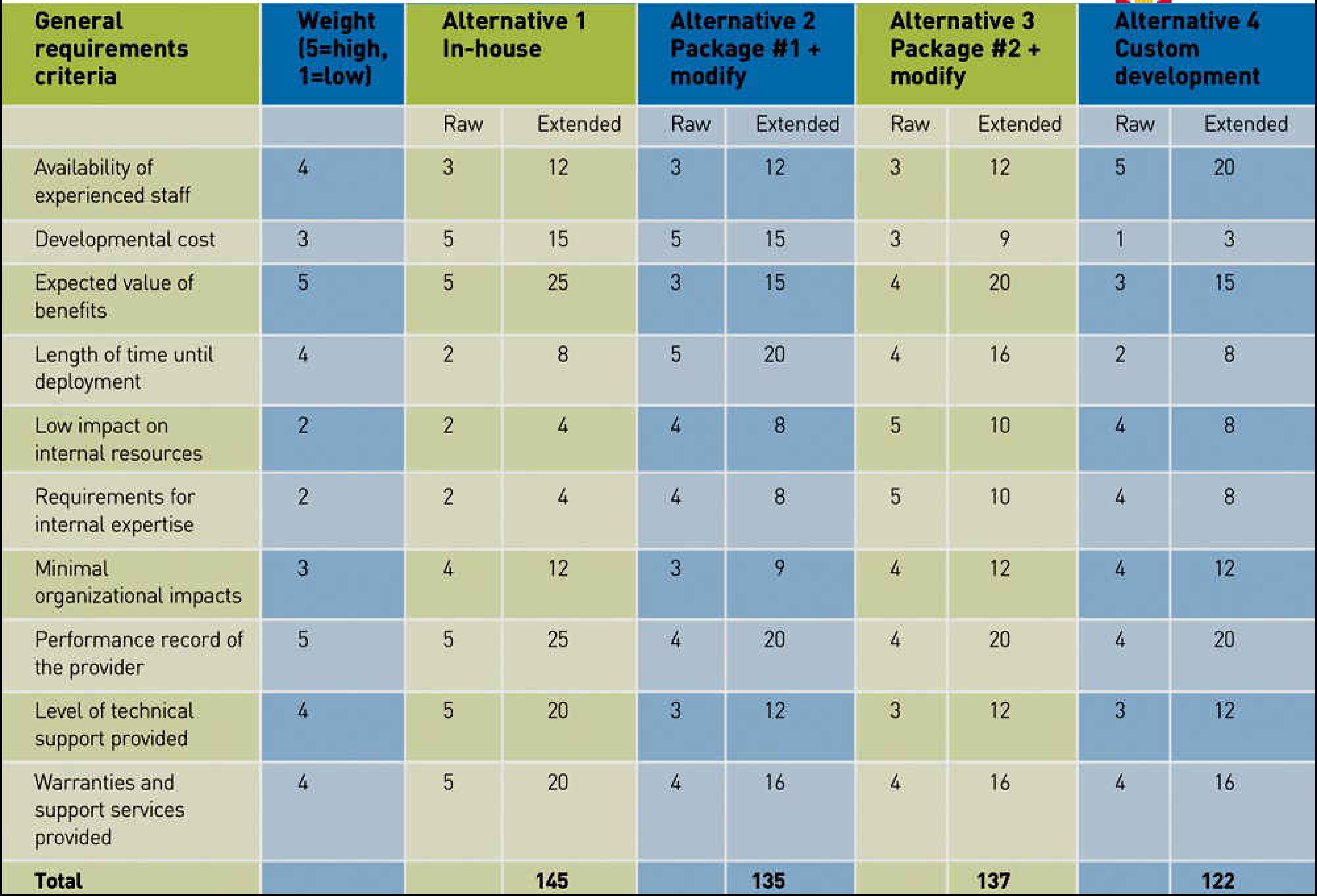
  Comparisons can be difficult as different proposed systems have strengths in different areas

- See example on next slides - 3 major areas considered:
    - ➢ General requirements
    - ➢ Technical requirements
    - ➢ Functional requirements

1. Each alternative rated
2. Weighted scores are then tabulated and compared to make a choice

Image from Satzinger, J. Jackson, R. & Burd, S. (2004) Systems Analysis and Design in a Changing World, 3rd edition. Course Technology, Thomson Learning. Figure 8.7

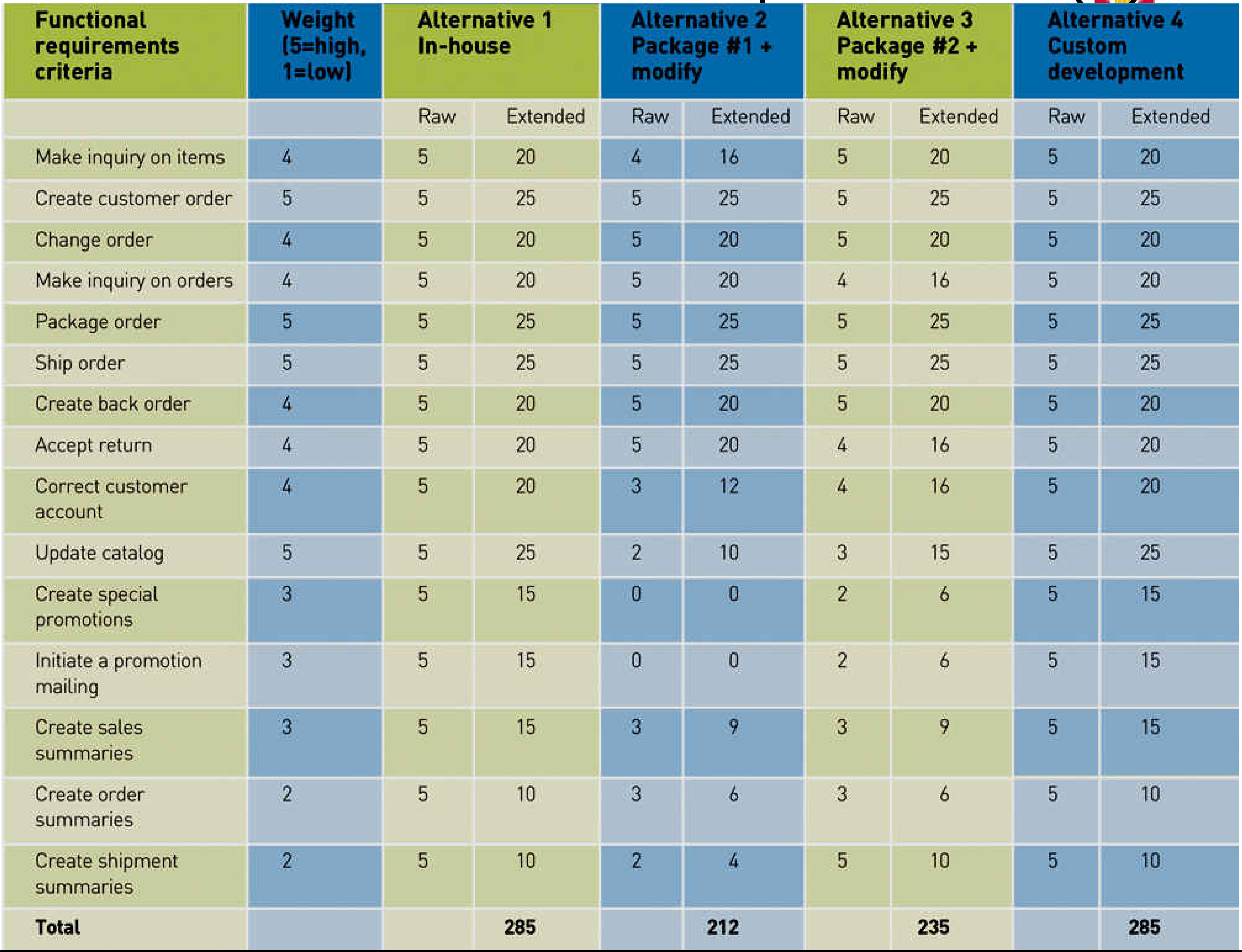
Partial matrix of general requirements (1)



General requirements criteria	Weight (5=high, 1=low)	Alternative 1 In-house		Alternative 2 Package #1 + modify		Alternative 3 Package #2 + modify		Alternative 4 Custom development	
		Raw	Extended	Raw	Extended	Raw	Extended	Raw	Extended
Availability of experienced staff	4	3	12	3	12	3	12	5	20
Developmental cost	3	5	15	5	15	3	9	1	3
Expected value of benefits	5	5	25	3	15	4	20	3	15
Length of time until deployment	4	2	8	5	20	4	16	2	8
Low impact on internal resources	2	2	4	4	8	5	10	4	8
Requirements for internal expertise	2	2	4	4	8	5	10	4	8
Minimal organizational impacts	3	4	12	3	9	4	12	4	12
Performance record of the provider	5	5	25	4	20	4	20	4	20
Level of technical support provided	4	5	20	3	12	3	12	3	12
Warranties and support services provided	4	5	20	4	16	4	16	4	16
Total			**145**		**135**		**137**		**122**

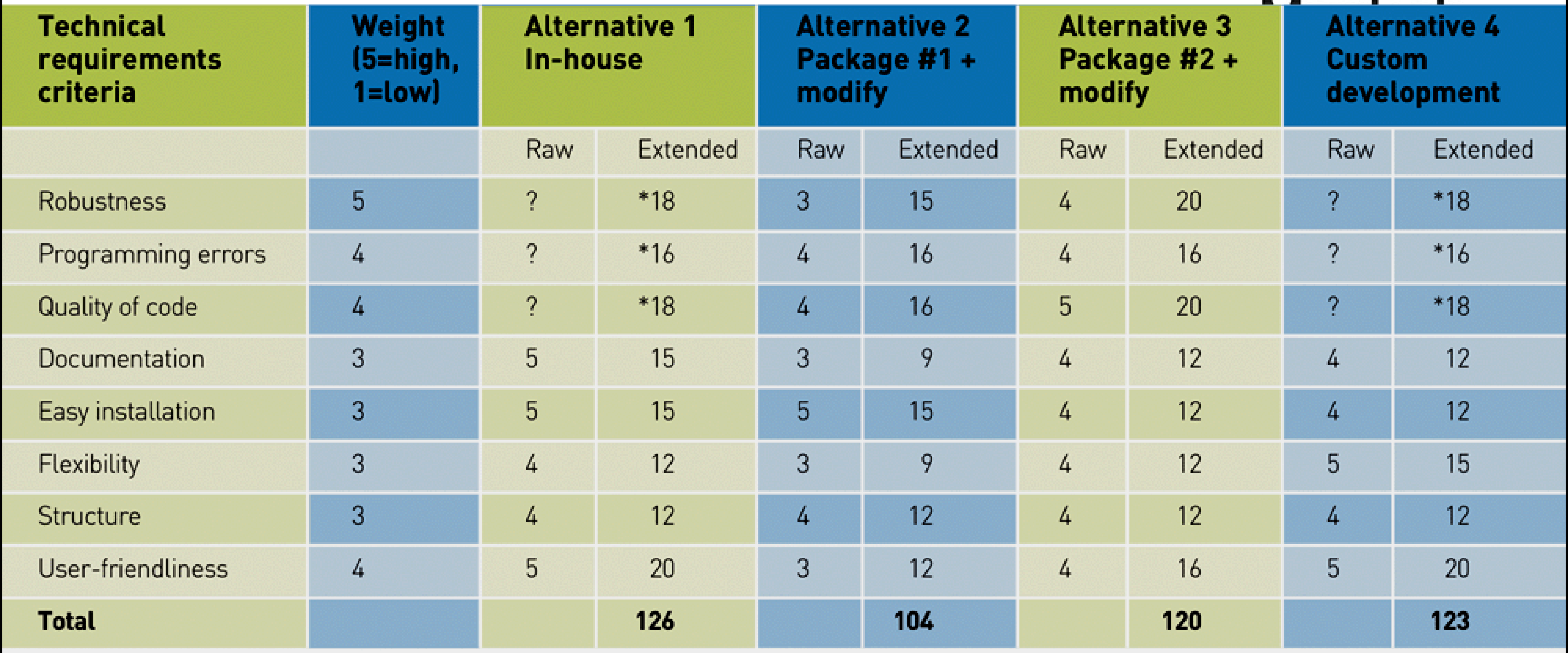
Image from Satzinger, J. Jackson, R. & Burd, S. (2004) Systems Analysis and Design in a Changing World, 3rd edition. Course Technology, Thomson Learning. Figure 8.8

Partial matrix of functional requirements (2)



Functional requirements criteria	Weight (5=high, 1=low)	Alternative 1 In-house		Alternative 2 Package #1 + modify		Alternative 3 Package #2 + modify		Alternative 4 Custom development	
		Raw	Extended	Raw	Extended	Raw	Extended	Raw	Extended
Make inquiry on items	4	5	20	4	16	5	20	5	20
Create customer order	5	5	25	5	25	5	25	5	25
Change order	4	5	20	5	20	5	20	5	20
Make inquiry on orders	4	5	20	5	20	4	16	5	20
Package order	5	5	25	5	25	5	25	5	25
Ship order	5	5	25	5	25	5	25	5	25
Create back order	4	5	20	5	20	5	20	5	20
Accept return	4	5	20	5	20	4	16	5	20
Correct customer account	4	5	20	3	12	4	16	5	20
Update catalog	5	5	25	2	10	3	15	5	25
Create special promotions	3	5	15	0	0	2	6	5	15
Initiate a promotion mailing	3	5	15	0	0	2	6	5	15
Create sales summaries	3	5	15	3	9	3	9	5	15
Create order summaries	2	5	10	3	6	3	6	5	10
Create shipment summaries	2	5	10	2	4	5	10	5	10
Total			**285**		**212**		**235**		**285**

Partial matrix of technical requirements (3)



Technical requirements criteria	Weight (5=high, 1=low)	Alternative 1 In-house		Alternative 2 Package #1 + modify		Alternative 3 Package #2 + modify		Alternative 4 Custom development	
		Raw	Extended	Raw	Extended	Raw	Extended	Raw	Extended
Robustness	5	?	*18	3	15	4	20	?	*18
Programming errors	4	?	*16	4	16	4	16	?	*16
Quality of code	4	?	*18	4	16	5	20	?	*18
Documentation	3	5	15	3	9	4	12	4	12
Easy installation	3	5	15	5	15	4	12	4	12
Flexibility	3	4	12	3	9	4	12	5	15
Structure	3	4	12	4	12	4	12	4	12
User-friendliness	4	5	20	3	12	4	16	5	20
Total			**126**		**104**		**120**		**123**

Total score:

Alternative 1 = 556
Alternative 2 = 451
Alternative 3 = 475
Alternative 4 = 530

Image from Satzinger, J. Jackson, R. & Burd, S. (2004) Systems Analysis and Design in a Changing World, 3rd edition. Course Technology, Thomson Learning. Figure 8.9

Summing up…

- **Feasibility** is the measure of how beneficial or practical the development of an information system will be to an organisation
  - Organisational
  - Technical
  - Resource and economic
  - Schedule feasibility
  - Legal feasibility
- Feasibility should be measured and re-evaluated throughout the system development life cycle (*creeping commitment* approach)
- A **feasibility matrix** is a useful way to compare the feasibility of different solutions, e.g. in the ‘buy v build’ decision

Designing for security and integrity

Designing for security and integrity

Security and integrity controls must be considered in the design of EVERY other element we have discussed

- not a stand-alone activity

- Integrity controls
  - Maintaining the integrity (correctness) of the data as it is entered into the system, processed, stored and output
- Security controls
  - Associated with the entire environment, primarily (but not always) external threats, that are often malicious

Designing for security and integrity

Controls are needed for all design aspects:

- User interface – limit access to authorized users
- System interface – protect from other systems
- Application architecture – record transactions
- Database – protect from software/hardware failure
- Network design – protect communications

Integrity and security controls

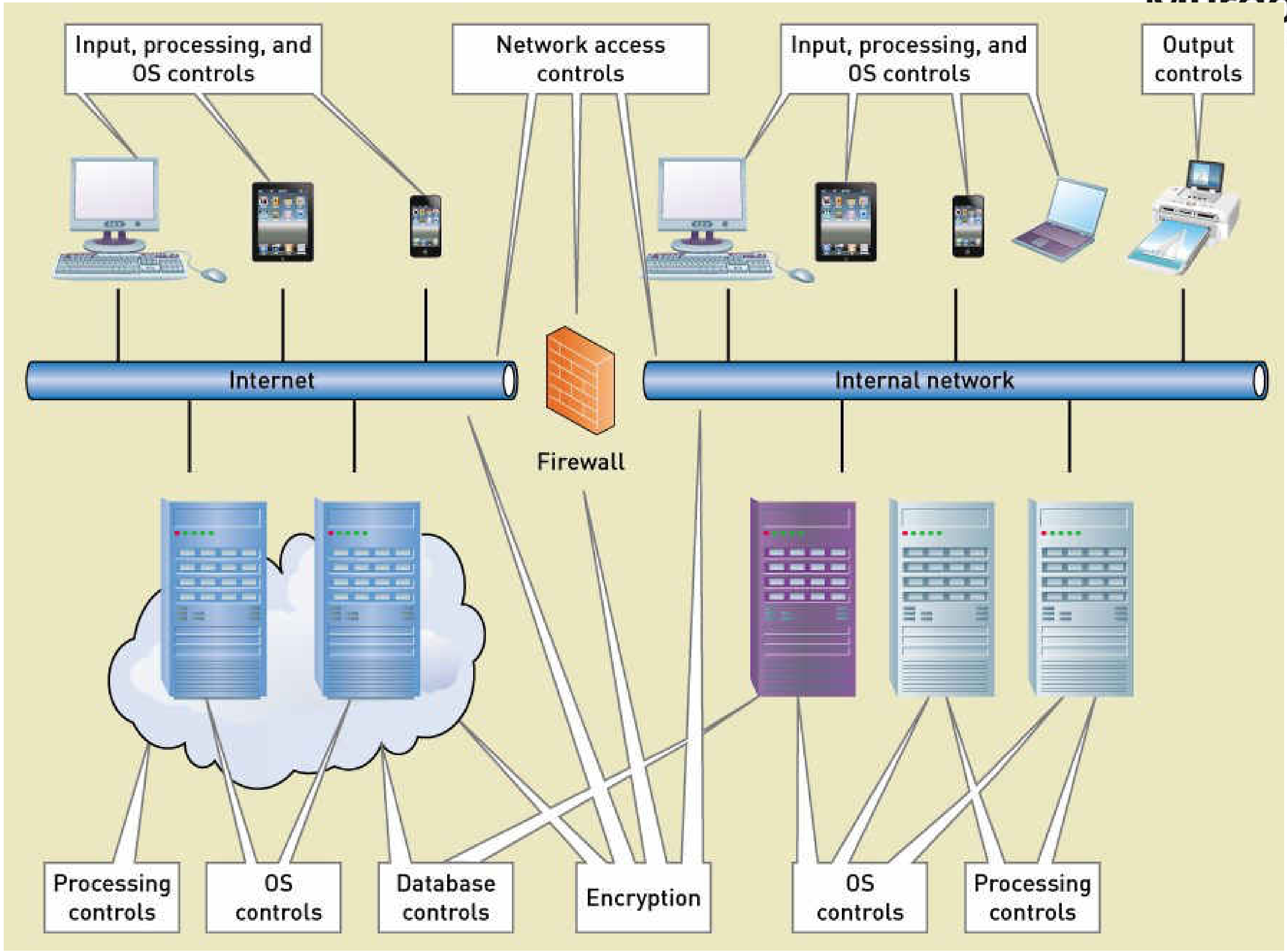


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Designing integrity controls

- Objectives of integrity controls:
  - Ensure that only appropriate and correct business transactions are accepted
  - Ensure that transactions are recorded and processed correctly
  - Protect and safeguard assets such as the database
- Integrated into application programs and DBMS:
  - Input controls
  - Output controls
  - Backup and recovery
  - Fraud prevention (next slides)

Input controls

Prevent invalid or erroneous data from entering the system

- *Value Limit Controls*

  Check the range of inputs for reasonableness

- *Completeness Controls*

  Ensure all the data has been entered

- *Data Validation Controls*

  Ensure that specific data values are correct

- *Field Combination Controls*

  Ensure data is correct based on relationships between fields

Output controls

To ensure that output arrives at proper destination (for authorized eyes) and is accurate, current, and complete

Controls include:

- Physical access to printers and display devices
- Discarded data – protect from “dumpster diving”
- Access controls to programs that display or print
- Formatting and labelling printed outputs to show they are date stamped
- Labelling of electronic output

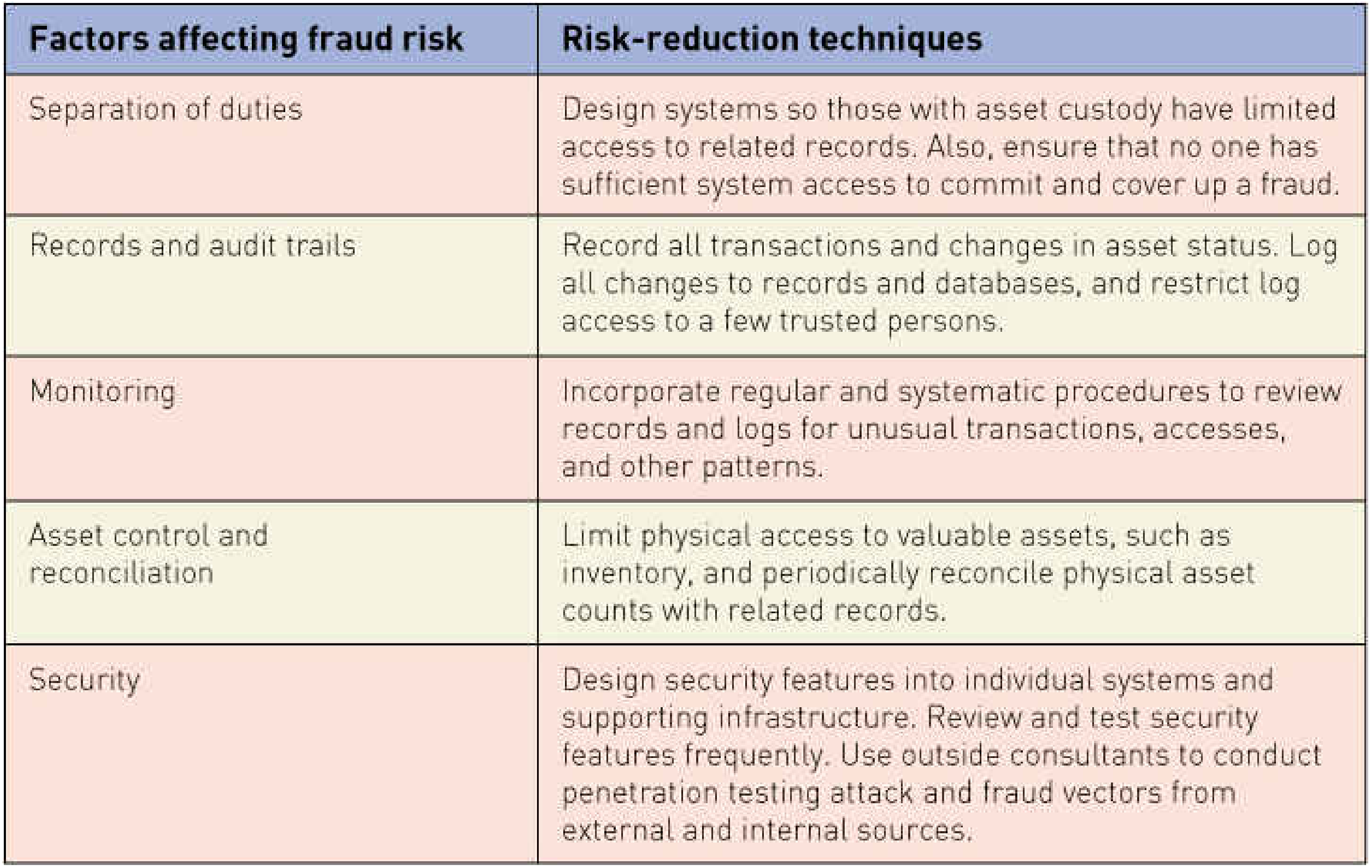
Redundancy, backup and recovery

Protect data and systems from catastrophes

- Databases
- Hardware
- Software applications
- Networks

- Redundant servers, database, sites
- Backup and recovery
- On-site versus off-site copies

Fraud prevention – risk factors and risk reduction techniques



Factors affecting fraud risk	Risk-reduction techniques
Separation of duties	Design systems so those with asset custody have limited access to related records. Also, ensure that no one has sufficient system access to commit and cover up a fraud.
Records and audit trails	Record all transactions and changes in asset status. Log all changes to records and databases, and restrict log access to a few trusted persons.
Monitoring	Incorporate regular and systematic procedures to review records and logs for unusual transactions, accesses, and other patterns.
Asset control and reconciliation	Limit physical access to valuable assets, such as inventory, and periodically reconcile physical asset counts with related records.
Security	Design security features into individual systems and supporting infrastructure. Review and test security features frequently. Use outside consultants to conduct penetration testing attack and fraud vectors from external and internal sources.

Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Designing security controls

- Protect all assets against external threats
- Protect and maintain a stable, functioning operating environment 24/7 (equipment, operating systems, DBMSs)
- Protect information and transactions during transmission across networks and Internet
- Approaches:

  Access controls

  Data encryption

  Digital signatures and certificates

  Secure transactions

Designing security controls – user access

- Access Controls – Limit a person’s ability to access servers, files, data, applications
  - Authentication – to identify users
  - Access control list – list of valid users and their permissions for a resource
  - Authorization – the process of allowing or restricting access for a specific user
- Registered Users – those with authorization
- Unauthorized Users – those not allowed access
- Privileged Users – those that maintain lists and systems

Types of users

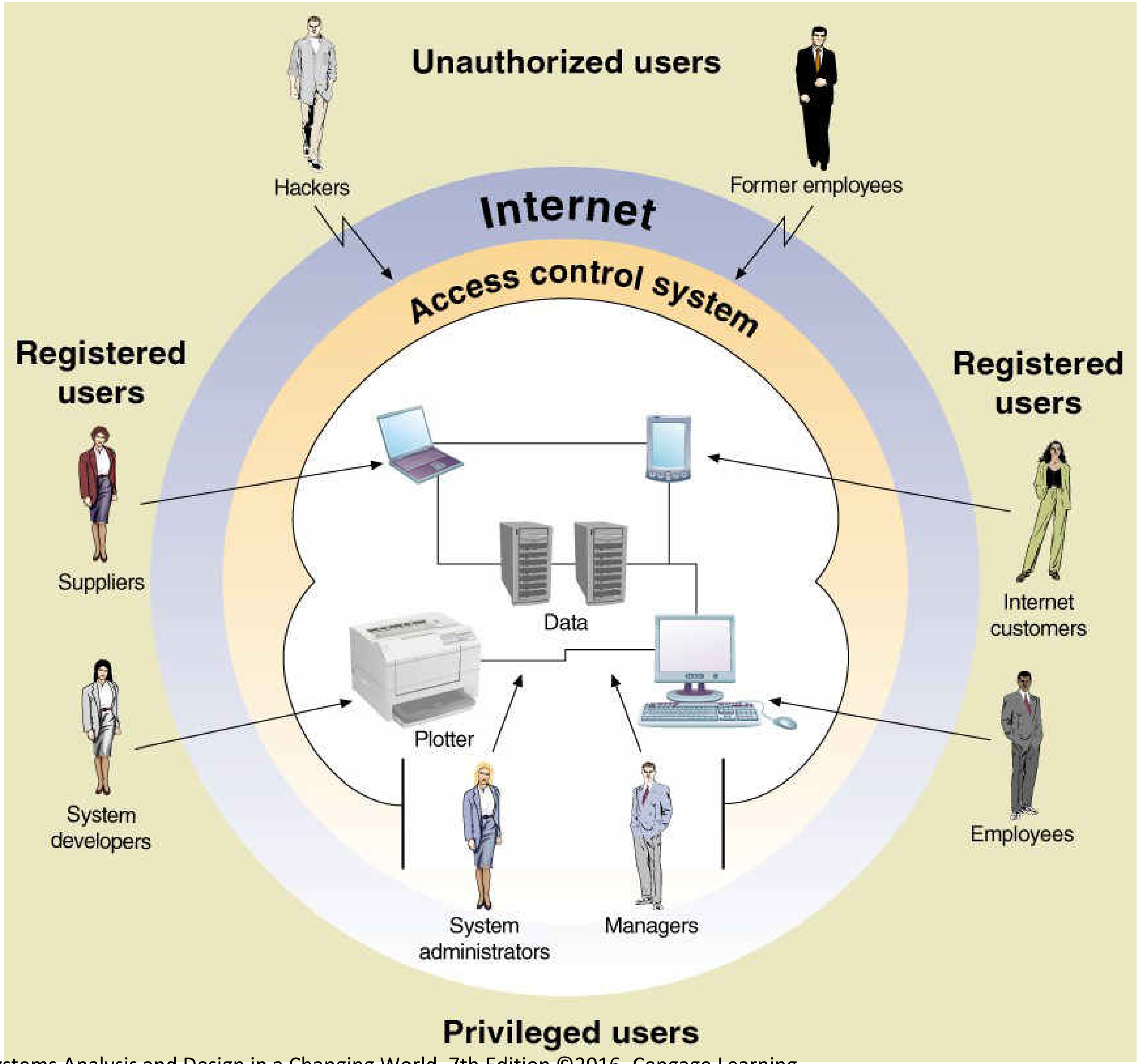


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Designing security controls – Encryption, digital signatures and certificates

- Encryption is the primary method of securing data within internal systems and during transmission
    - Encryption alters data/files so that unauthorised users can’t view them
    - **Keys** are used to encrypt and decrypt
- Digital signatures and digital certificates are a way of verifying identity so that keys can be used by legitimate parties

Designing security controls – Secure transactions

- Secure transactions use a standard set of methods and protocols that address authentication, authorisation, privacy and integrity
  - Transport Layer Security (TLS) or Secure Sockets Layer (SSL)
  - IP Security (IPSec)
  - HTTPS (Hypertext Transfer Protocol Secure)

Summing up…

- **Integrity controls** maintain the integrity (correctness) of the data as it is entered into the system, processed, stored and output
    - Input controls
    - Output controls
    - Backup and recovery
    - Fraud prevention
- **Security controls** are associated with the entire environment, primarily (but not always) external threats, that are often malicious
    - Access controls
    - Data encryption
    - Digital signatures and certificates
    - Secure transactions

Topic learning outcomes revisited

After completing this topic you should be able to:

- Describe systems design and contrast it with systems analysis
- List the documents and models used as inputs to or output from systems design
- Describe each major design activity
- Describe how different alternative solutions may be evaluated for feasibility
- Discuss the advantages and disadvantages involved in the 'buy versus build' decision for software components or applications
- Explain why security and integrity methods and controls need to be considered throughout all design activities
- Briefly describe some security and integrity methods and controls

Murdoch UNIVERSITY

What’s next?

In the next topic, we’ll commence our coverage of systems design activities by looking at the high level systems architecture, and designing the database.