Object Oriented Analysis & Design

Week 12

Object Oriented Design and Project Management

Lecturer: Dr. Msagha J Mbogholi, PhD

Flashback from Lesson 11

- The OO principle of modularity makes code more reusable as functionality is divided across different methods.
- Cohesion is a measure of the degree to which elements of a module are related. In OO context cohesion is examined in the context of classes, objects and methods (these are the individual parts of the module).
- A good software design will have high cohesion
- Coupling simply refers to the degree of interdependence or interrelation of modules in a system; consequently, little or no coupling is desirable.
- Connascence literally means to be born together. From an object-oriented design perspective, it really means that two modules (classes or methods) are so intertwined that if you make a change in one, it is likely that a change in the other will be required. Consequently, connascence should be minimized.
- Reuse refers to using components of one product to develop a different product with different functionality. There are two types of reuse – opportunistic and systematic.
- Design patterns are divided into three categories: creational, structural and behavioral.
- The key problems involving portability revolve around different compilers, operating systems, and hardware configurations.

Content

- Introduction
- Project identification
- Feasibility analysis
- Traditional project management tools
- Workplan creation and management
- IEEE software project management plan

Part 1

Introduction

Introduction

- We are at the last lesson for this course. Hopefully though, this is just the beginning for you.
- Throughout the course you have learnt the phases of the SDLC and more specifically in the object oriented domain.
- What remains is for you to work the phases and implement your own solution to a problem domain.
- Unfortunately many organizations get stuck precisely at this point since they do not have the requisite skills and/or knowledge to manage the process.
- The development of the system should be approached as a project; preferably using the rigors of project management.
- In this last lesson we go through the stages of project management using the Unified Process(UP) and how to apply them in our domain, namely OOAD.
- In the final part of the lesson we examine the IEEE software project management plan in fair detail.

Introduction (cont'd)

- Projects are an everyday part of life should you care to think about it. We even use the term "project" to refer to personal activities which we are currently undertaking, or which we are planning to undertake in the future.
- A project is simply a series of tasks undertaken within a specific time frame with the purpose of achieving a certain goal. You may say you're undertaking a project to build a house; you will normally have a plan on how to do this with a specific time frame and a breakdown of the activities necessary to complete the project. The smart thing to do is to give yourself timeframes during which specific components of the project will be complete; for example, foundation to be done within six months, and so on.
- *Project management* is the process of planning and controlling the development of a system within a specified time frame at a minimum cost with the right functionality. (Dennis et al, 2015)

Introduction (cont'd)

- Project management has become such an important process in business that we now have its study as an independent profession. Thus a project manager (the one who manages the project) will be someone with project management skills and not necessarily from an IT background. This person can manage a project in any discipline, be it IT, engineering, finance, or even manufacturing. They do this using a set series of steps in the management and delivery of the final product.
- The field of IT is ever changing; new technology is being developed literally on a daily basis. This may encourage an organization to want to be at par with it, as today's world competition is not something any organization would wish to ignore. In fact in some industries the failure to incorporate new technology can cost an organization its competitive edge (where the cheese has moved; have you read the book?)
- However, the bigger question is whether newer is better in all circumstances?
- In the field of information technology, whenever there is a new product there are key issues that need to be examined before incorporating it into the business. First and foremost of course is whether there is even a budget for it in the first place! Thereafter, other issues can be examined like whether the organization really needs it, what is the value added, and so on. So clearly some organized approach to this is required.
- Enter project management, the discipline.

Introduction (cont'd)

- When developing a system for the organization there are many things that may go wrong in the design and implementation phase, resulting in a maintenance nightmare for the developer (and organization by extension).
- This is why the unified process (UP) was developed in the first place; to give the whole process a systematic and disciplined approach.
- In system development and this lesson we examine project management using the UP. As a reminder the phases of the UP are inception, elaboration, construction and transition. The details in each phase have been covered throughout this course.
- In the inception phase of the UP of a new systems development project a business values that can be gained from an information technology is identified (this can be done by a business analyst or technical person). At this stage it is important to note that the business value should be clearly seen whether it is in the form of tangible or intangible benefits. More importantly the system should clearly align with the business objectives of the organization; added value in terms of increased sales and so on, will interest and put the sponsors (whether it is top management or directors) squarely on board. This will trigger the rest of the UP phases. Let us summarize what happens before describing the details in the remaining sections of this lesson.

Introduction (cont'd)

- Dennis et al (2015) describe the general process as follows:"
- To ensure that a real business need is being addressed, the affected business organization (called the *project sponsor*), proposes the new systems development project using a *system request.* (this officially kicks off the inception phase of the UP)
- The request is forwarded to an *approval committee* for consideration. The approval committee reviews the request and makes an initial determination of whether to investigate the proposal or not. If the committee initially approves the request, the systems development team gathers more information to determine the feasibility of the project.
- A *feasibility analysis* plays an important role in deciding whether to proceed with an information systems development project. It examines the technical, economic, and organizational pros and cons of developing the system, and it gives the organization a slightly more detailed picture of the advantages of investing in the system as well as any obstacles that could arise.
- Once the feasibility analysis has been completed, it is submitted to the approval committee, along with a revised system request. The committee then decides whether to approve the project, decline the project, or table it until additional information is available.

Introduction (cont'd)

- Projects are selected by weighing risks and returns and by making trade-offs at the organizational level."
- Should the committee approve the system request the following will happen (Dennis et al, 2015):
- Because we are following a Unified Process-based approach, the systems development workplan will evolve throughout the development process.
- Given this evolutionary approach, one critical success factor for project management is to start with a realistic assessment of the work that needs to be accomplished and then manage the project according to that assessment.
- This can be achieved by carefully creating and managing the workplan, estimating the eff ort to develop the system, staffing the project, and coordinating project activities."

Part 2

Project identification

2.1 Project identification

- Projects are identified when someone in the organization realizes there is a need for a system.
- This someone could be a technical or business staff. Usually it is advisable that it comes from a business staff such as a business analyst as these will clearly state the business need of the department or organization. A few key points regarding project identification:
- Source – the initiator can be a department or it can be a need across the organization. If it is from a department then the initiator may be a business analyst or a technical staff. If it is across the organization then it should be a business analyst.
- Need – the business need can be based on new technology or it may be an improvement to existing technology. If it is based on a new system then it will preferably involve both business and technical staff. If it is an improvement this can be handled by technical staff as the initial analysis (for the existing system) had already been approved).

2.1 Project identification (cont'd)

- Benefit – the sponsor needs to be appraised of the benefits of the system from a business perspective. Dennis et al (2015) describe it as follows: "The business need drives the high-level *business requirements* for the system. Requirements are what the information system will do, or the *functionality* it will contain. They need to be explained at a high level so that the approval committee and, ultimately, the project team understand what the business expects from the final product. Business requirements are the features and capabilities the information system will have to include, such as the ability to collect customer orders online or the ability for suppliers to receive inventory information as orders are placed and sales are made."
- Value – the project sponsor also needs to be told what the business value of the system to be developed is to the organization. Values can be either tangible or intangible. Tangible values are those which can be seen or quantified; for example, a 5% increase in sales, or a 20% reduction in production costs. Intangible values are those which can't be quantified but can be felt; for example, a boost in employee morale (which in turn leads to a boost in employee productivity, which is tangible), or better customer service (which again will in turn result in an expected increase in sales), and so on.

2.1 Project identification (cont'd)

- Scope - The *project sponsor* is someone who recognizes the strong business need for a system and has an interest in seeing the system succeed. Depending on the size and scope of the project the sponsor can range from a unit manager to the organization CEO. The scope actually determines the kind of sponsor required. If the system (project) is for a department then the sponsor can be a unit manager; however, if it is across the organization then top level manager (or even CEO) should be the sponsor. The sponsor can also be the IT department (especially in instances where the project is technical, such as improvements to existing infrastructure), and in some circumstances it can be both IT and business departments.
- Once the sponsor has identified the business need and the value to be gained from the project, s/he can formally initiate the project. This is done via a document called the system request.

2.2 System request

- A system request is a document that describes the business reasons for building a system and the value that the system is expected to provide. The project sponsor usually completes this form as part of a formal system project selection process within the organization. (Dennis et al, 2015)
- The system request contains five elements:
  - Project sponsor – the name of the project sponsor
  - Business need – a description of the business need being fulfilled
  - Business requirements – the requirements of the business being addressed by this request.
  - Business value – what is the business value of the requested system (tangible/intangible)
  - Special issues – any special issues that require consideration.
- The system is then submitted to an approval committee for vetting. The committee then determines whether the project requires further investigation or not (in which case they will either shelve it or reject it). Should they determine that it requires further investigation the project moves to the next phase, which is the feasibility study.

Part 3

Feasibility analysis

Introduction

- A feasibility analysis allows the organization to determine whether to go ahead with the project or not.
- This analysis examines areas where there is risk, whether the organization has the necessary skills and equipment to carry out the project, and whether it is truly feasible, that is whether it is truly possible and convenient.
- In practice feasibility analysis is done throughout the project lifecycle. However, at this stage an initial feasibility analysis is done. It is done using three feasibility areas:
  - Technical feasibility
  - Economic feasibility
  - Organizational feasibility

3.1 Technical feasibility

- Technical feasibility refers to "the extent to which the system can be successfully designed, developed, and installed by the IT group." (Dennis et al, 2015)
- In a nutshell technical feasibility asks the question "can we build it?". This means taking into consideration capabilities as well as risks involved in carrying out the project.
- Risks – some of the risks that need consideration include:
- The analyst's lack of familiarity with the functional area (functionality) of the proposed system. This may make the analyst not understand the users or even identify opportunities for improvement.
- Familiarity with technology. If users and technical staff aren't familiar with the technology more time will be used training them, and it may still result in problems that weren't anticipated.
- Project size should also be considered a risk; the bigger the project the harder it is to manage and also the more attention to detail is required.
- Compatibility with existing systems is another risk that needs to be considered.
- In some organizations it may also make sense to involve an outsider to help in assessing technical feasibility.

3.1 Technical feasibility

- Technical feasibility can thus be summarized as follows (Dennis et al, 2015):
- **Technical Feasibility: Can We Build It?**
- Familiarity with Functional area: Less familiarity generates more risk
- Familiarity with Technology: Less familiarity generates more risk
- Project Size: Large projects have more risk
- Compatibility: The harder it is to integrate the system with the company's existing technology, the higher the risk

3.2 Economic feasibility

- Economic feasibility is really about performing cost-benefit analysis (CBA).
- This means weighing out the costs of developing the system versus the perceived benefits the system will provide.
- Economic feasibility is determined by identifying costs and benefits associated with the system, assigning values to them, and then calculating the cash flow and return on investment for the project. The more expensive the project, the more rigorous and detailed the analysis should be.
- The steps taken in performing an economic feasibility are provided by Dennis et al (2015) as follows:

3.2 Economic feasibility (cont'd)

- **Step 1. Identifying Costs and Benefits -** List the tangible costs and benefits for the project. Include both one-time and recurring costs.
- **Step 2. Assigning Values to Costs and Benefits -** Work with business users and IT professionals to create numbers for each of the costs and benefits. Even intangibles should be valued if at all possible.
- **Step 3. Determining Cash Flow -** Project what the costs and benefits will be over a period of time, usually three to five years. Apply a growth rate to the numbers, if necessary.
- **Step 4. Determining Net Present Value (NPV) -** Calculate what the value of future costs and benefits are if measured by today's standards. You will need to select a rate of growth to apply the NPV formula.
- **Step 5. Determining Return on Investment (ROI) -** Calculate how much money the organization will receive in return for the investment it will make using the ROI formula.

3.2 Economic feasibility (cont'd)

- **Step 6. Determining the Break-Even Point -** Find the first year in which the system has greater benefits than costs. Apply the break-even formula using figures from that year. This will help you understand how long it will take before the system creates real value for the organization.
- **Step 7. Graphing the Break-Even Point -** Plot the yearly costs and benefits on a line graph. The point at which the lines cross is the break-even point.
- The details on how to perform the individual calculations can be found in Dennis et al (2015).
- However, fig 1 shows an example of a cost benefit analysis. Fig 2 assigns values to he costs and benefits in fig 1, while fig 3 shows the projected cash flows for the example over a five year period (costs and benefits over a number of years)

Development Costs	Operational Costs
Development Team Salaries	Software Upgrades
Consultant Fees	Software Licensing Fees
Development Training	Hardware Repairs
Hardware and Software	Hardware Upgrades
Vendor Installation	Operational Team Salaries
Office Space and Equipment	Communications Charges
Data Conversion Costs	User Training
Tangible Benefits	**Intangible Benefits**
Increased Sales	Increased Market Share
Reductions in Staff	Increased Brand Recognition
Reductions in Inventory	Higher Quality Products
Reductions in IT Costs	Improved Customer Service
Better Supplier Prices	Better Supplier Relations

Fig 1. Cost benefit analysis (Dennis et al, 2015)

Benefits[a]	
Increased sales	500,000
Improved customer service[b]	70,000
Reduced inventory costs	68,000
Total benefits	**638,000**
Development costs	
2 servers @ $125,000	250,000
Printer	100,000
Software licenses	34,825
Server software	10,945
Development labor	1,236,525
Total development costs	**1,632,295**
Operational costs	
Hardware	54,000
Software	20,000
Operational labor	111,788
Total operational costs	**185,788**
Total costs	**1,818,083**

[a] An important yet intangible benefit will be the ability to offer services that our competitors currently offer.

[b] Customer service numbers have been based on reduced costs for customer complaint phone calls.

Fig 2. Assigning values to costs and benefits (Dennis et al, 2015)

	2015	2016	2017	2018	2019	Total
Increased sales	500,000	530,000	561,800	595,508	631,238	
Reduction in customer complaint calls	70,000	70,000	70,000	70,000	70,000	
Reduced inventory costs	68,000	68,000	68,000	68,000	68,000	
TOTAL BENEFITS:	638,000	668,000	699,800	733,508	769,238	
PV OF BENEFITS:	**619,417**	**629,654**	**640,416**	**651,712**	**663,552**	**3,204,752**
PV OF ALL BENEFITS:	**619,417**	**1,249,072**	**1,889,488**	**2,541,200**	**3,204,752**	
2 Servers @ $125,000	250,000	0	0	0	0	
Printer	100,000	0	0	0	0	
Software licenses	34,825	0	0	0	0	
Server software	10,945	0	0	0	0	
Development labor	1,236,525	0	0	0	0	
TOTAL DEVELOPMENT COSTS:	1,632,295	0	0	0	0	
Hardware	54,000	81,261	81,261	81,261	81,261	
Software	20,000	20,000	20,000	20,000	20,000	
Operational labor	111,788	116,260	120,910	125,746	130,776	
TOTAL OPERATIONAL COSTS:	185,788	217,521	222,171	227,007	232,037	
TOTAL COSTS:	1,818,083	217,521	222,171	227,007	232,037	
PV OF COSTS:	**1,765,129**	**205,034**	**203,318**	**201,693**	**200,157**	**2,575,331**
PV OF ALL COSTS:	**1,765,129**	**1,970,163**	**2,173,481**	**2,375,174**	**2,575,331**	
TOTAL PROJECT BENEFITS COSTS:	**(1,180,083)**	**450,479**	**477,629**	**506,501**	**537,201**	
YEARLY NPV:	**(1,145,712)**	**424,620**	**437,098**	**450,019**	**463,395**	**629,421**
CUMULATIVE NPV:	**(1,145,712)**	**(721,091)**	**(283,993)**	**166,026**	**629,421**	
RETURN ON INVESTMENT:	**24.44%**	(629,421/2,575,331)				
BREAK-EVEN POINT:	**3.63 years**	[break-even occurs in year 4; (450,019 − 166,026)/450,019 = 0.63]				
INTANGIBLE BENEFITS:	This service is currently provided by competitors Improved customer satisfaction					

Fig 3. Cash flow analysis (CBA). (Dennis et al, 2015)

3.3 Organizational feasibility

- Organizational feasibility assesses how well the project will be received by the users.
- The users who will use the system or are affected by it are referred to as stakeholders. A lot of times systems are developed which are ultimately not used by the very users they were designed for.
- This is a challenge because users will resist using a system for various reasons; for example, they don't see how it adds value, they believe (and sometimes rightly so) that it interferes with or makes their work harder, or simply that they weren't involved in the system development process.
- One way to go around this is to develop systems that will align with the business objectives or strategy of the organization or department. This is where business analysts are so important to the process.
- A stakeholder analysis can also be done where all the stakeholders will be involved in the process of making the system acceptable for use.
- There are three key stakeholders involved: the champion, organizational management, and the users.

Their roles are explored in fig 4.

	Role	Techniques for Improvement
Champion	A champion: • Initiates the project • Promotes the project • Allocates his or her time to project • Provides resources	• Make a presentation about the objectives of the project and the proposed benefits to those executives who will benefit directly from the system • Create a prototype of the system to demonstrate its potential value
Organizational Management	Organizational managers: • Know about the project • Budget enough money for the project • Encourage users to accept and use the system	• Make a presentation to management about the objectives of the project and the proposed benefits • Market the benefits of the system using memos and organizational newsletters • Encourage the champion to talk about the project with his or her peers
System Users	Users: • Make decisions that influence the project • Perform hands-on activities for the project • Ultimately determine whether the project is successful by using or not using the system	• Assign users official roles on the project team • Assign users specific tasks to perform with clear deadlines • Ask for regular feedback from users (e.g., at weekly meetings)

Fig 4. Stakeholders for organizational feasibility (Dennis et al, 2015)

3.4 Project selection

- The feasibility study together with the system request (including any revisions that may have been done in the interim) are then submitted to the approval committee.
- The committee will assess the feasibility juxtaposed with the system request in order to determine the way forward. The committee may give either of the following decisions:
  - Approve project
  - Reject project
  - Hold project (awaiting further information)
- The decision of the committee is based on examining the business needs against the risk involved.
- Further the decision is also based on the organization's portfolio; that is, the different projects that exist in the organization, big or small, high or low risk, tactical or strategic. This is called portfolio management.
- Fig 5 shows different ways of classifying projects in a portfolio.

Size	What is the size? How many people are needed to work on the project?
Cost	How much will the project cost the organization?
Purpose	What is the purpose of the project? Is it meant to improve the technical infrastructure? Support a current business strategy? Improve operations? Demonstrate a new innovation?
Length	How long will the project take before completion? How much time will go by before value is delivered to the business?
Risk	How likely is it that the project will succeed or fail?
Scope	How much of the organization is affected by the system? A department? A division? The entire corporation?
Return on investment	How much money does the organization expect to receive in return for the amount the project costs?

Fig 5. Project classification in portfolio (Dennis et al, 2015)

Part 4

Traditional project management tools

Introduction

- Once a project has been approved the next task is make the project manageable. This is done by breaking down the project into tasks and deliverables.
- A task is a unit of work carried out to achieve a target; it is carried out by a person or a team.
- A deliverable is a tangible outcome of a task or a group of related tasks; it can be a document or a complete module.
- The project manager will oversee the completion of all tasks that will cumulatively result in the completion of the project. The task is the smallest unit of work in the project and is described as an action. A task will have a name, description, a start and end date, person(s) assigned to it, deliverable(s), completion status, priority (high, low, normal), and any resources required to complete it. Fig 6 shows an example of task information.
- In order to assist in managing the project, the project manager will use project management tools that will assist in tracking tasks and deliverables, as well as provide details of the different phases and tasks in the project. The tools also show how different tasks are related.

4.1 Tools

- There are three main tools used in project management:
- Work breakdown structure (WBS) – this tool takes tasks, breaks them down into subtasks (where need be), show their dependencies on other tasks (if any), status (whether complete or in progress), and their duration. Fig 7 shows an example of a WBS.
- Gantt chart – this tool shows what is on the WBS but in a graphical way (a picture is worth a thousand words right?) so that that the breakdown of tasks and assignment can be easily visualized. Fig 8 shows an example of a Gantt chart.
- Network diagram – this tool shows the tasks in a flowchart. PERT (project evaluation and review technique) are used when individual task time estimates aren't certain. PERT uses three time estimates: optimistic, most likely, and pessimistic; it then uses a weighted average formula to get a time estimate. Fig 9 shows an example of a network diagram.

Workplan Information	Example
Name of the task	Perform economic feasibility
Start date	Jan 05, 2015
Completion date	Jan 19, 2015
Person assigned to the task	Project sponsor: Mary Smith
Deliverable(s)	Cost-benefit analysis
Completion status	Open
Priority	High
Resources that are needed	Spreadsheet software
Estimated time	16 hours
Actual time	14.5 hours

Fig 6. Task information (Dennis et al, 2015)

Task Number	Task Name	Duration (in weeks)	Dependency	Status
1	Identify vendors	2		Complete
2	Review training materials	6	1	Complete
3	Compare vendors	2	2	In Progress
4	Negotiate with vendors	3	3	Open
5	Develop communications information	4	1	In Progress
6	Disseminate information	2	5	Open
7	Create and administer survey	4	6	Open
7.1	Create initial survey	1		Open
7.2	Review initial survey	1	7.1	Open
7.2.1	Review by Director of IT Training	1		Open
7.2.2	Review by Project Sponsor	1		Open
7.2.3	Review by Representative Trainee	1		Open
7.3	Pilot test initial survey	1	7.1	Open
7.4	Incorporate survey changes	1	7.2, 7.3	Open
7.5	Create distribution list	0.5		Open
7.6	Send survey to distribution list	0.5	7.4, 7.5	Open
7.7	Send follow-up message	0.5	7.6	Open
7.8	Collect completed surveys	1	7.6	Open
8	Analyze results and choose vendor	2	4, 7	Open
9	Build new classrooms	11	1	In Progress
10	Develop course options	3	8, 9	Open

Fig 7. Work breakdown structure (Dennis et al, 2015)

ID	Task Name	Duration	Start	Finish	Prede
1	Identify vendors	2 wks	Wed 1/1/15	Tue 1/14/15	
2	Review training materials	6 wks	Wed 1/1/15	Tue 2/11/15	
3	Compare vendors	2 wks	Wed 2/12/15	Tue 2/25/15	2
4	Negotiate with vendors	3 wks	Wed 2/26/15	Tue 3/8/15	3
5	Develop communications information	4 wks	Wed 1/15/15	Tue 2/11/15	1
6	Disseminate information	2 wks	Wed 2/12/15	Tue 2/25/15	5
7	Create and administer survey	4 wks	Wed 2/26/15	Tue 3/25/15	6
8	Analyze results and choose	2 wks	Wed 3/26/15	Tue 4/8/15	4, 7
9	Build new classroom	11 wks	Wed 1/15/15	Tue 4/1/15	1
10	Develop course options	3 wks	Wed 4/9/15	Tue 4/29/15	8, 9
11	Budget Meeting	1 day	Wed 1/15/15	Wed 1/15/15	
12	Software Installation	1 day	Tue 4/1/15	Tue 4/1/15	

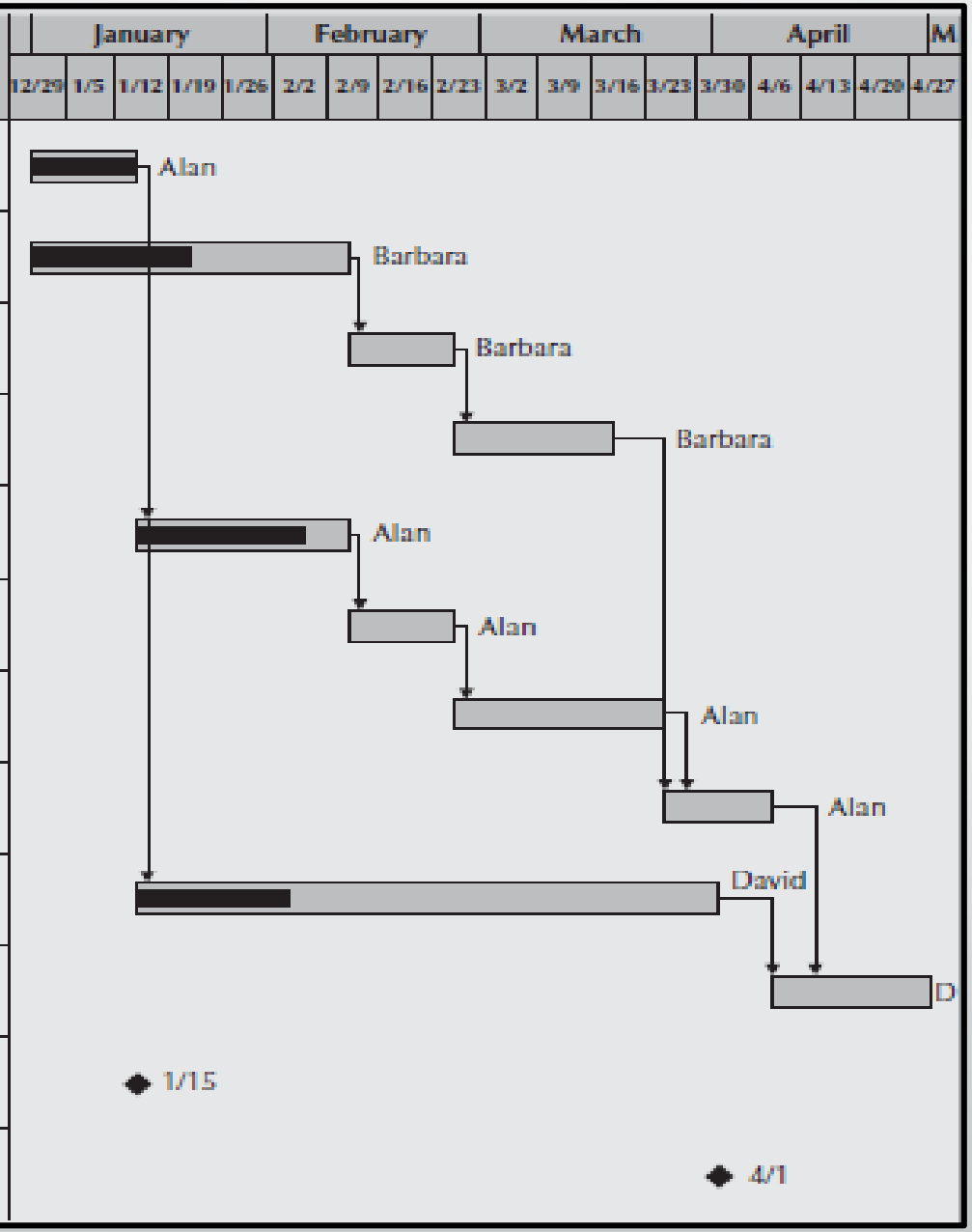

Fig 8. Gantt chart (Dennis et al, 2015)

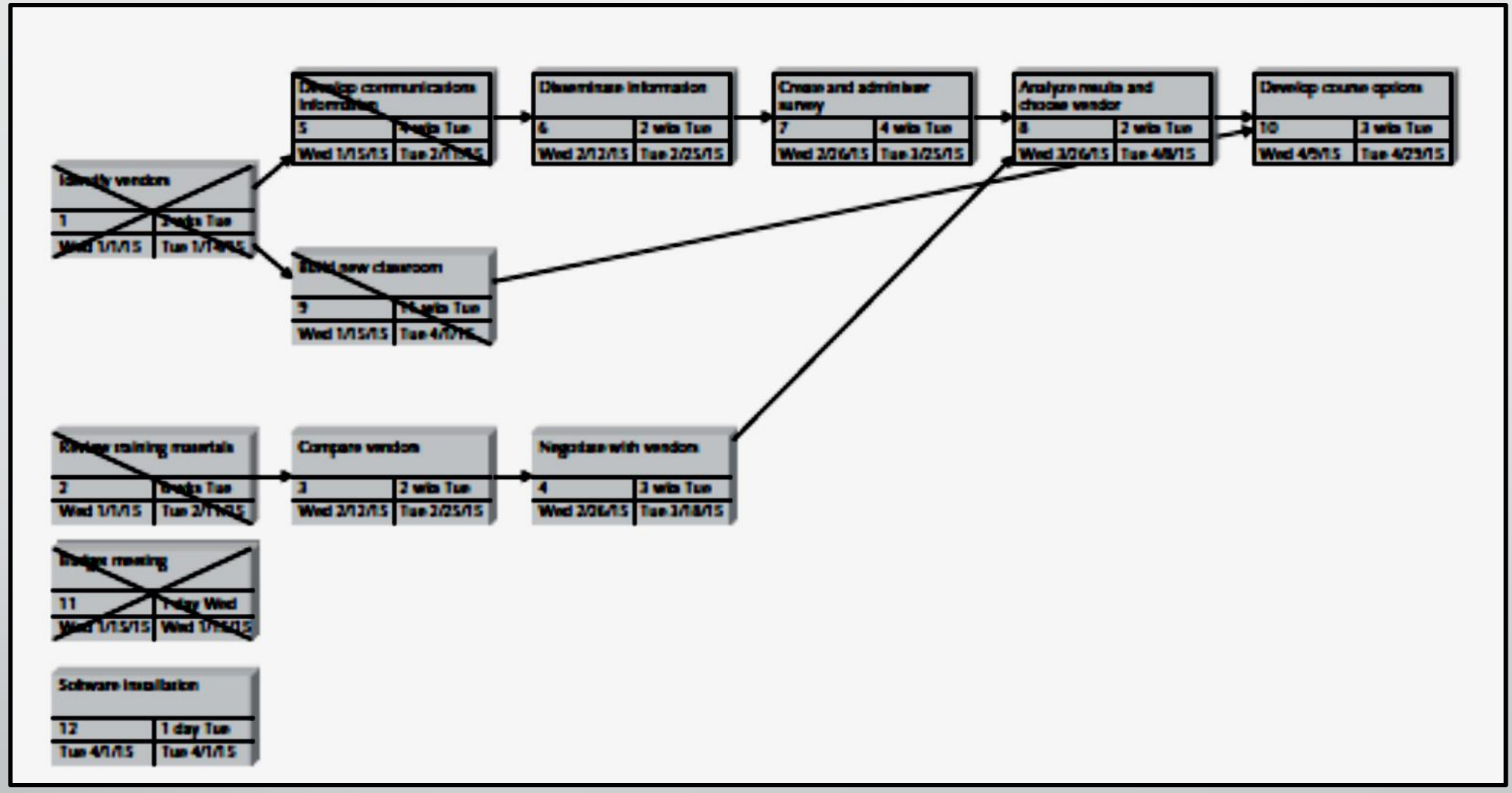

Fig 9. Network diagram (Dennis et al, 2015)

Part 5

Workplan creation and management

Introduction

- The workplan "is a dynamic schedule that records and keeps track of all the tasks that need to be accomplished over the course of the project. The workplan lists each task, along with important information about it, such as when it needs to be completed, the person assigned to do the work, and any deliverables that will result. The level of detail and the amount of information captured by the workplan depend on the needs of the project, and the detail usually increases as the project progresses". (Dennis et al, 2015)
- The key question is how to build tasks for a system that you have never developed before? Well the answer to this is simple if you are building an object oriented system.
- There are many different methodologies available in software engineering and the project team can pick one. The best way to do this is to study methodologies that have been used to build systems similar to the one being built. Secondly an approach to the development is then determined. The UP as we know is based on an iterative and incremental approach.
- With this information each phase of the project is broken down into the UP steps and consequent iterations performed. This can be used to create an evolutionary WBS as shown in fig 10.

I. Business Modeling
   a. Inception
   b. Elaboration
   c. Construction
   d. Transition
   e. Production

II. Requirements
   a. Inception
   b. Elaboration
   c. Construction
   d. Transition
   e. Production

III. Analysis
   a. Inception
   b. Elaboration
   c. Construction
   d. Transition
   e. Production

IV. Design
   a. Inception
   b. Elaboration
   c. Construction
   d. Transition
   e. Production

V. Implementation
   a. Inception
   b. Elaboration
   c. Construction
   d. Transition
   e. Production

VI. Test
   a. Inception
   b. Elaboration
   c. Construction
   d. Transition
   e. Production

VII. Deployment
   a. Inception
   b. Elaboration
   c. Construction
   d. Transition
   e. Production

VIII. Configuration and Change Management
   a. Inception
   b. Elaboration
   c. Construction
   d. Transition
   e. Production

IX. Project Management
   a. Inception
   b. Elaboration
   c. Construction
   d. Transition
   e. Production

X. Environment
   a. Inception
   b. Elaboration
   c. Construction
   d. Transition
   e. Production

XI. Operations and Support
   a. Inception
   b. Elaboration
   c. Construction
   d. Transition
   e. Production

XII. Infrastructure Management
   a. Inception
   b. Elaboration
   c. Construction
   d. Transition
   e. Production

Fig 10. Evolutionary WBS template for enhanced UP (Dennis et al, 2015)

5.1 Other management tasks

- Managing scope - Scope creep happens when new requirements are added to the project after the original project scope was defined and frozen. However, using an iterative and incremental development process allows the team to deal with changing requirements in an effective way. A couple of useful agile techniques to manage the scope of the project while attempting to satisfy the client are daily scrum meetings and the product backlog used with Scrum. (Dennis et al, 2015)
- Another way of managing scope is timeboxing. This technique sets a fixed deadline for a project and delivers the system by that deadline no matter what, even if functionality needs to be reduced. Timeboxing ensures that project teams don't get hung up on the final finishing touches that can drag out indefinitely, and it satisfies the business by providing a product within a relatively short time frame. (Dennis et al, 2015)
- The two remaining management tasks are :
- Refine estimates as the project progresses; this is because as the project progresses what was unknown becomes clearer and thus estimates can now be refined (in the time and cost dimensions).
- Risk management which is anchored on risk assessment; the risk assessment takes into account all the potential risks in the project and their impact and likelihood of occurring. You may read more on risk assessment in the computer network security course (lesson 4) by yours truly (https://www.hufocw.org/Course/858)

5.2 Remaining project tasks

- There will be only two remaining tasks once this stage is complete:
- Task 1 - Project staffing. This will involve selection of the project team in terms of what each staff will be required to do (based on their skillset) and constitution of an effective project team. The team can be split in terms of functional and technical components, while the manager will be expected to provide leadership and motivation.
- Task 2 - Environment and infrastructure management. The environment workflow primarily deals with choosing the correct set of tools that will be used throughout the development process and identifying the appropriate set of standards to be followed during the development process. Infrastructure management workflow deals with choosing the appropriate level and type of documentation that will be created during the development process. Other activities associated with the infrastructure management workflow include developing, modifying, and reusing predefined components, frameworks, libraries, and patterns. (Dennis et al, 2015)
- Details of these two tasks can be found in Dennis et al (2015).

Part 6

IEEE software project management plan

Introduction

- The IEEE standard 1058-1998 describes the format and content of a software management plan.
- The components of the plan are shown in fig 11.
- Schach (2017) provides an explanation of the contents and this is captured in the following slides.

1 Overview
- 1.1 Project summary
  - 1.1.1 Purpose, scope, and objectives
  - 1.1.2 Assumptions and constraints
  - 1.1.3 Project deliverables
  - 1.1.4 Schedule and budget summary
- 1.2 Evolution of the project management plan

2 Reference materials

3 Definitions and acronyms

4 Project organization
- 4.1 External interfaces
- 4.2 Internal structure
- 4.3 Roles and responsibilities

5 Managerial process plans
- 5.1 Start-up plan
  - 5.1.1 Estimation plan
  - 5.1.2 Staffing plan
  - 5.1.3 Resource acquisition plan
  - 5.1.4 Project staff training plan
- 5.2 Work plan
  - 5.2.1 Work activities
  - 5.2.2 Schedule allocation
  - 5.2.3 Resource allocation
  - 5.2.4 Budget allocation
- 5.3 Control plan
  - 5.3.1 Requirements control plan
  - 5.3.2 Schedule control plan
  - 5.3.3 Budget control plan
  - 5.3.4 Quality control plan
  - 5.3.5 Reporting plan
  - 5.3.6 Metrics collection plan
- 5.4 Risk management plan
- 5.5 Project close-out plan

6 Technical process plans
- 6.1 Process model
- 6.2 Methods, tools, and techniques
- 6.3 Infrastructure plan
- 6.4 Product acceptance plan

7 Supporting process plans
- 7.1 Configuration management plan
- 7.2 Testing plan
- 7.3 Documentation plan
- 7.4 Quality assurance plan
- 7.5 Reviews and audits plan
- 7.6 Problem resolution plan
- 7.7 Subcontractor management plan
- 7.8 Process improvement plan

8 Additional plans

Fig 11. IEEE project management plan framework (Schach, 2017)

6.1 SPMP framework

- The IEEE software project management plan (SPMP) framework has been provided in fig 11. Schach (2017) provides an explanation for the contents:"
- **1 Overview.**
- **1.1 Project summary.**
- **1.1.1 Purpose, scope, and objectives.** A brief description is given of the purpose and scope of the software product to be delivered, as well as project objectives. Business needs are included in this subsection.
- **1.1.2 Assumptions and constraints.** Any assumptions underlying the project are stated here, together with constraints, such as the delivery date, budget, resources, and artifacts to be reused.
- **1.1.3 Project deliverables.** All the items to be delivered to the client are listed here, together with the delivery dates.
- **1.1.4 Schedule and budget summary.** The overall schedule is presented here, together with the overall budget.

6.1 SPMP framework

- **1.2 Evolution of the project management plan.** No plan can be cast in concrete. The project management plan, like any other plan, requires continual updating in the light of experience and change within both the client organization and the software development organization. In this section, the formal procedures and mechanisms for changing the plan are described, including the mechanism for placing the project management plan itself under configuration control.
- **2 Reference materials.** All documents referenced in the project management plan are listed here.
- **3 Definitions and acronyms.** This information ensures that the project management plan will be understood the same way by everyone.
- **4 Project organization.**
- **4.1 External interfaces.** No project is constructed in a vacuum. The project members have to interact with the client organization and other members of their own organization. In addition, subcontractors may be involved in a large project. Administrative and managerial boundaries between the project and these other entities must be laid down.

6.1 SPMP framework

- **4.2 Internal structure.** In this section, the structure of the development organization itself is described. For example, many software development organizations are divided into two types of groups: development groups that work on a single project and support groups that provide support functions, such as configuration management and quality assurance, on an organization-wide basis. Administrative and managerial boundaries between the project group and the support groups also must be defined clearly.
- **4.3 Roles and responsibilities.** For each project function, such as quality assurance,
- and for each activity, such as product testing, the individual responsible must be identifi ed.
- **5 Managerial process plans.**
- **5.1 Start-up plan.**
- **5.1.1 Estimation plan.** The techniques used to estimate project duration and cost are listed here, as well as the way these estimates are tracked and, if necessary, modified while the project is in progress.
- **5.1.2 Staffing plan.** The numbers and types of personnel required are listed, together with the durations for which they are needed.

6.1 SPMP framework

- **5.1.3 Resource acquisition plan.** The way of acquiring the necessary resources, including hardware, software, service contracts, and administrative services, is given here.
- **5.1.4 Project staff training plan.** All training needed for successful completion of the project is listed in this subsection.
- **5.2 Work plan.**
- **5.2.1 Work activities.** In this subsection, the work activities are specified, down to the task level if appropriate.
- **5.2.2 Schedule allocation.** In general, the work packages are interdependent and further dependent on external events. For example, the implementation workflow follows the design workflow and precedes product testing. In this subsection, the relevant dependencies are specified.
- **5.2.3 Resource allocation.** The various resources previously listed are allocated to the appropriate project functions, activities, and tasks.
- **5.2.4 Budget allocation.** In this subsection, the overall budget is broken down at the project function, activity, and task levels.

6.1 SPMP framework

- **5.3 Control plan.**
- **5.3.1 Requirements control plan.** As described in Part B of this book, while a software product is being developed, the requirements frequently change. The mechanisms used to monitor and control the changes to the requirements are given in this section.
- **5.3.2 Schedule control plan.** In this subsection, mechanisms for measuring progress are listed, together with a description of the actions to be taken if actual progress lags behind planned progress.
- **5.3.3 Budget control plan.** It is important that spending should not exceed the budgeted amount. Control mechanisms for monitoring when actual cost exceeds budgeted cost, as well as the actions to be taken should this happen, are described in this subsection.
- **5.3.4 Quality control plan.** The ways in which quality is measured and controlled are described in this subsection.
- **5.3.5 Reporting plan.** To monitor the requirements, schedule, budget, and quality, reporting mechanisms need to be in place. These mechanisms are described in this subsection.

6.1 SPMP framework

- **5.3.6 Metrics collection plan.** As explained in Section 5.5, it is not possible to
- manage the development process without measuring relevant metrics. The metrics to be collected are listed in this subsection.
- **5.4 Risk management plan.** Risks have to be identified, prioritized, mitigated, and tracked. All aspects of risk management are described in this section.
- **5.5 Project close-out plan.** The actions to be taken once the project is completed, including reassignment of staff and archiving of artifacts, are presented here.
- **6 Technical process plans.**
- **6.1 Process model.** In this section, a detailed description is given of the life-cycle model to be used.
- **6.2 Methods, tools, and techniques.** The development methodologies and programming languages to be used are described here.

6.1 SPMP framework

- **6.3 Infrastructure plan.** Technical aspects of hardware and software are described in detail in this section. Items that should be covered include the computing systems (hardware, operating systems, network, and software) to be used for developing the software product, as well as the target computing systems on which the software product will be run and CASE tools to be employed.
- **6.4 Product acceptance plan.** To ensure that the completed software product passes its acceptance test, acceptance criteria must be drawn up, the client must agree to the criteria in writing, and the developers must then ensure that these criteria are indeed met. The way that these three stages of the acceptance process will be carried out is described in this section.
- **7 Supporting process plans.**
- **7.1 Configuration management plan.** In this section, a detailed description is given of the means by which all artifacts are put under configuration management.

6.1 SPMP framework

- **7.2 Testing plan.** Testing, like all other aspects of software development, needs careful planning.
- **7.3 Documentation plan.** A description of documentation of all kinds, whether or not to be delivered to the client at the end of the project, is included in this section.
- **7.4 Quality assurance plan.** All aspects of quality assurance, including testing, standards, and reviews, are encompassed by this section.
- **7.5 Reviews and audits plan.** Details as to how reviews are conducted are presented in this section.
- **7.6 Problem resolution plan.** In the course of developing a software product, problems are all but certain to arise. For example, a design review may bring to light a critical fault in the analysis workflow that requires major changes to almost all the artifacts already completed. In this section, the way such problems are handled is described.

6.1 SPMP framework

- **7.7 Subcontractor management plan.** This section is applicable when subcontractors are to supply certain work products. The approach to selecting and managing subcontractors then appears here.
- **7.8 Process improvement plan.** Process improvement strategies are included in this section.
- **8 Additional plans.** For certain projects, additional components may need to appear in the plan. In terms of the IEEE framework, they appear at the end of the plan. Additional components may include security plans, safety plans, data conversion plans, installation plans, and the software project postdelivery maintenance plan.

Summary

- *Project management* is the process of planning and controlling the development of a system within a specified time frame at a minimum cost with the right functionality.
- The *project sponsor* is someone who recognizes the strong business need for a system and has an interest in seeing the system succeed. Depending on the size and scope of the project the sponsor can range from a unit manager to the organization CEO.
- A system request is a document that describes the business reasons for building a system and the value that the system is expected to provide. The project sponsor usually completes this form as part of a formal system project selection process within the organization.
- Feasibility analysis is done using technical feasibility, economic feasibility and organizational feasibility.
- Traditional project management tools include work breakdown structure (WBS), Gantt charts and network diagrams.
- The IEEE standard 1058-1998 describes the format and content of a software management plan.

References

- Dennis, A., Wixom, B. H., Tegarden, D. P., & Seeman, E. (2015). *System analysis & design: An object-oriented approach with Uml*. Wiley.
- Schach, S. R. (2017). *Object-oriented and classical software engineering*. Langara College.