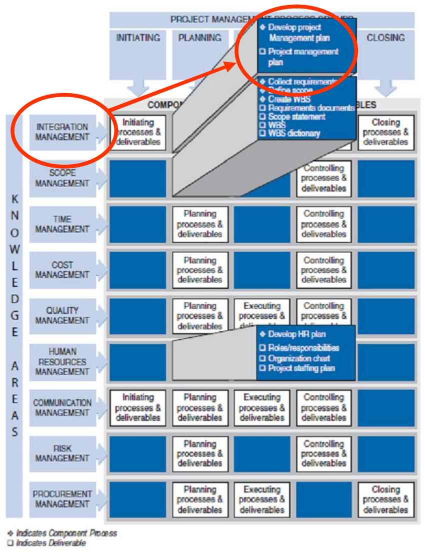
Chapter 5

Project Planning. Project Scope. Work Breakdown Structure.

Objectives

- Acquire a general understanding of the parts of the project management plan
- Understand the importance of discovering and documenting stakeholder requirements
- Understand how to create a detailed scope statement and WBS
- Learn how to match the right person, with the needed skill set, to the appropriate activity

Project Planning. Integration Management KA.



Project Planning starts with the Project Plan Development process, which is a part of the Integration Management knowledge area.

The single deliverable from this process is the Project Management Plan – a consistent coherent document; it consists of deliverables for each of the other 8 Knowledge Areas (KAs).

Project Plan =
Telling the team "WHAT TO DO"

Project Management Plan: Main Components

▪ Scope management plan	(Chapter 5)	Lab 1
▪ Work breakdown structure (WBS)	(Chapter 5)	Lab 1
▪ Human Resource management plan	(Chapter 5)	Lab 1
▪ Time (schedule) management plan	(Chapter 6)	Lab 2
▪ Cost management plan	(Chapter 6)	Lab 2
▪ Quality management plan	(Chapter 7)	
▪ Process improvement plan	(Chapter 7)	
▪ Communication management plan	(Chapter 7)	
▪ Risk management plan	(Chapter 8)	Lab 3
▪ Procurement management plan	(Chapter 9)	

Many organizations not only have documented templates for each part of the plan to speed up development of the plan and to maintain consistency across projects but also may have slightly different standard formats, based on different project characteristics, such as size, complexity, length, and risk level.

For example, a small, low-risk project would have a shorter and less formal project planning document than a large, complex project with members of the project team spread out all over the world.

Attributes of Good Project Plans

- Plans should be dynamic
- Plans should be flexible

  Things always happen during the project to change each of these four constraints so plans should be built to accommodate room for issues/problems/delays.

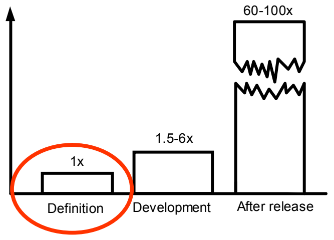
- Plans should be updated as changes occur (Integrated Change Control)
- Plans should first and foremost guide project execution
- Plans should ***never*** assume the team will work overtime, at least not at the start

  Assuming that the only way to hit the project plan objectives for 1) scope, 2) time, 3) cost, and 4) quality is to schedule over time at the very beginning is a sure recipe for disaster.

Why to Create Project Plan?

- The Cost of Software Change Law is a very well-known law (see on the right).
- Errors found "upstream" during the planning phase cost on the order of 200 times less to fix than errors found "downstream" during the building of the product.
- Planning "forecasting" "seeing into the future" is not an easy task
- T. Capers Jones (1998) summed it up this way: "The seeds of major software disasters are usually shown in the first three months of commencing the software project. Hasty scheduling, irrational commitments, unprofessional estimating techniques, carelessness of the project management function are the factors that tend to introduce terminal problems."

The Cost of SW Change



Project Plan Creation and *Analysis Paralysis*

- Although planning is crucial, project teams must be careful to **avoid over-planning**
- The project planning must be appropriate to the size, complexity, and risk of the project (small and easy projects -> small planning; large and complex projects -> significant planning efforts)
- Project managers must be careful to avoid what is known as "***analysis paralysis***" -- getting stuck in the analysis phase, trying to get everything defined perfectly

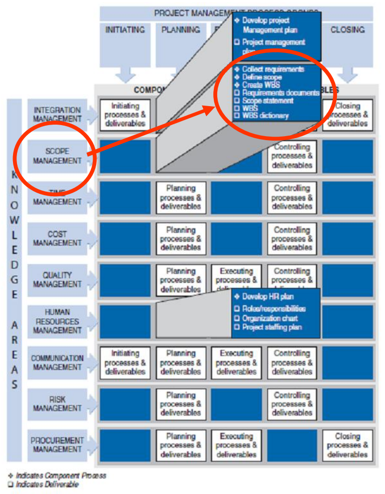
In software development, analysis paralysis typically manifests itself through exceedingly long phases of project planning, requirements gathering, program design and data modeling, with little or no extra value created by those steps. When extended over too long a timeframe, such processes tend to emphasize the organizational (i.e., bureaucratic) aspect of the software project, while detracting from its functional (value-creating) portion.

Analysis paralysis often occurs due to the lack of experience on the part of business systems analysts, project managers or software developers, as well as a rigid and formal organizational culture.

Analysis paralysis is an example of an anti-pattern. Agile software development methodologies explicitly seek to prevent analysis paralysis by promoting an iterative work cycle that emphasizes working products over product specifications.

CS590 "SE" vs. CS593 "SE of WebApp"courses

From Integration Management KA to Scope Management KA



Scope Management consists of 3 processes:

1. Collect requirements
2. Define Scope
3. Crete WBS (Work Breakdown Structure)

SW Requirements Engineering

A **requirement** is a singular documented need of what a particular product or service should be or perform. It is a statement that identifies a necessary **attribute, capability, function, characteristic, or quality of a system** in order for it to have value and utility to a user.

1) **Business requirements** describe in business terms ***WHAT*** must be delivered or accomplished to provide value.
2) **Product requirements** describe ***properties, functions and attributes*** of a system or product (which could be one of several ways to accomplish a set business requirements.)
3) **Process requirements** describe ***HOW activities performed by the developing organization*** (methodologies to be followed, and constraints that the organization must obey.

Main topics (or, components):

- Functional and nonfunctional system requirements
- Business rules (environment)
- Impacts on any other systems and/or departments
- Support and training requirements
- Acceptance criteria for each requirement or set of requirements
- Quality requirements

CS592 course

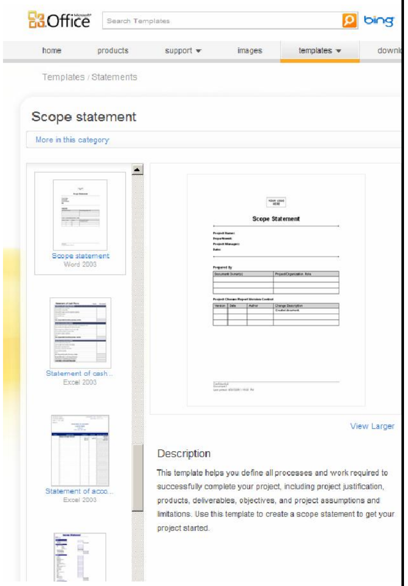
Scope Statement

Scope - refers to ***all (100%)*** of the work involved in creating the products of the project and the processes used to create them

A **scope statement** describes the characteristics of the product that the project was created to deliver.

Scope statements may take many forms depending on the type of project being implemented and the nature of the organization. However, a baseline scope statement should contain:

- The project name
- The project owner, sponsors, and stakeholders
- The project charter (roles and responsibilities, identities of stakeholders, etc.)
- The problem statement
- The project goals and objectives
- The project requirements
- The project deliverables
- The project non-goals (what is out of scope)
- Milestones (timetable, schedule)
- Cost estimates



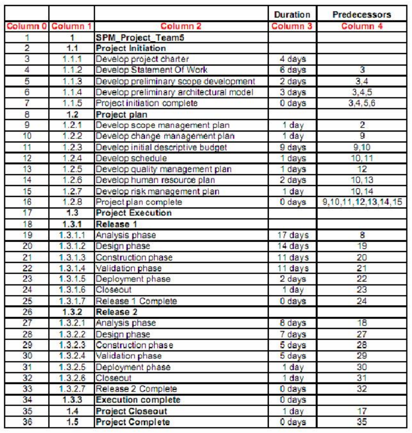
Description

This template helps you define all processes and work required to successfully complete your project, including project justification, products, deliverables, objectives, and project assumptions and limitations. Use this template to create a scope statement to get your project started.

Source: http://office.microsoft.com/en-us/templates/scope-statement-TC001142564.aspx

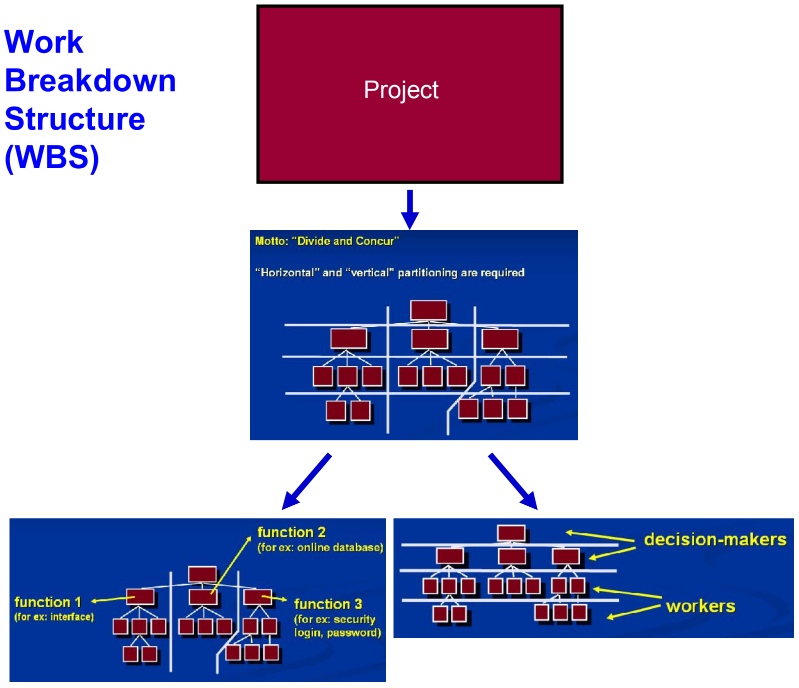
Work Breakdown Structure (WBS)

- A work breakdown structure (WBS) is a method used to define group of project's discrete work elements (or, tasks) in a way that helps organize and define the total work scope of the project.
- WBS element may be a
  - product,
  - data,
  - a component,
  - a service, or
  - any combination.
- 100% rule: The WBS represents 100 percent of the work required to produce the final products, and, therefore, all tasks must add up to 100% of the total scope and should not go over 100%.



			Duration	Predecessors
Column 0	Column 1	Column 2	Column 3	Column 4
1	**1**	**SPM_Project_Team5**		
2	**1.1**	**Project Initiation**		
3	1.1.1	Develop project charter	4 days	
4	1.1.2	Develop Statement Of Work	6 days	3
5	1.1.3	Develop preliminary scope development	2 days	3,4
6	1.1.4	Develop preliminary architectural model	3 days	3,4,5
7	1.1.5	Project initiation complete	0 days	3,4,5,6
8	**1.2**	**Project plan**		
9	1.2.1	Develop scope management plan	1 day	2
10	1.2.2	Develop change management plan	1 day	9
11	1.2.3	Develop initial descriptive budget	9 days	9,10
12	1.2.4	Develop schedule	1 days	10,11
13	1.2.5	Develop quality management plan	1 days	12
14	1.2.6	Develop human resource plan	2 days	10,13
15	1.2.7	Develop risk management plan	1 day	10,14
16	1.2.8	Project plan complete	0 days	9,10,11,12,13,14,15
17	**1.3**	**Project Execution**		
18	**1.3.1**	**Release 1**		
19	1.3.1.1	Analysis phase	17 days	8
20	1.3.1.2	Design phase	14 days	19
21	1.3.1.3	Construction phase	11 days	20
22	1.3.1.4	Validation phase	11 days	21
23	1.3.1.5	Deployment phase	2 days	22
24	1.3.1.6	Closeout	1 day	23
25	1.3.1.7	Release 1 Complete	0 days	24
26	**1.3.2**	**Release 2**		
27	1.3.2.1	Analysis phase	8 days	18
28	1.3.2.2	Design phase	7 days	27
29	1.3.2.3	Construction phase	5 days	28
30	1.3.2.4	Validation phase	5 days	29
31	1.3.2.5	Deployment phase	1 day	30
32	1.3.2.6	Closeout	1 day	31
33	1.3.2.7	Release 2 Complete	0 days	32
34	**1.3.3**	**Execution complete**	0 days	
35	**1.4**	**Project Closeout**	1 day	17
36	**1.5**	**Project Complete**	0 days	35

Work Breakdown Structure (WBS)


Project
Motto: "Divide and Concur"
"Horizontal" and "vertical" partitioning are required
function 2
(for ex: online database)
function 1
(for ex: interface)
function 3
(for ex: security login, password)
decision-makers
workers

Building the WBS

Various approaches can be used to build the WBS:

1. The **analogy approach**: A WBS is first created by looking for a similar projects done in the past and using its WBS as a starting point. SE Design Concept: “Do NOT reinvent the wheel” (check web sites of similar projects)

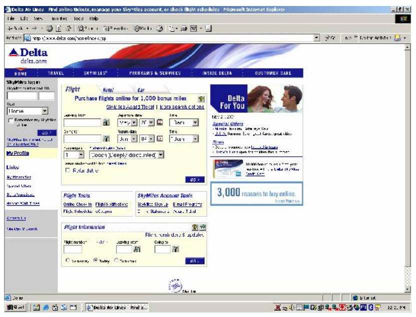
2. The **top-down approach**: Start with the largest items of the project and keep breaking them down into smaller and smaller parts

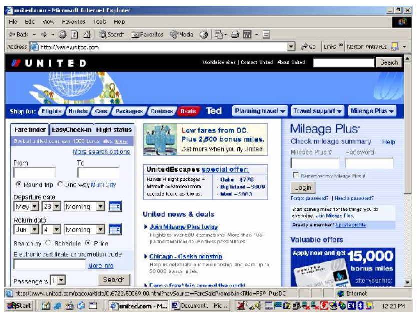
3. The **bottom-up approach**: Start with the detailed tasks and roll them up

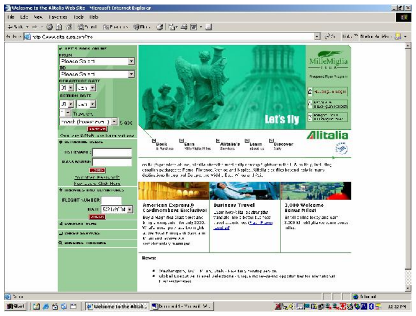
4. **Thread-based approach**: Concentrate on most important items first

*Using **guidelines**:*
Some organizations, like the DoD, National Science Foundation (NSF) provide guidelines/requirements for preparing a WBS

The analogy approach: a sample







Analogy Approach: Advantages and Issues

Advantages:

- Is the fastest path to a completed WBS
- Is a valuable tool for brainstorming a new project and looking for deliverables
- Enhances cross-project consistency
- Improves budget and time estimates
- Improves resource allocations

Issues:

- Ensure that the previous WBS is completely understood and similar
- Ensure the previous WBS is accurate and updated
- Critically review the previous WBS and its appropriateness for the new project

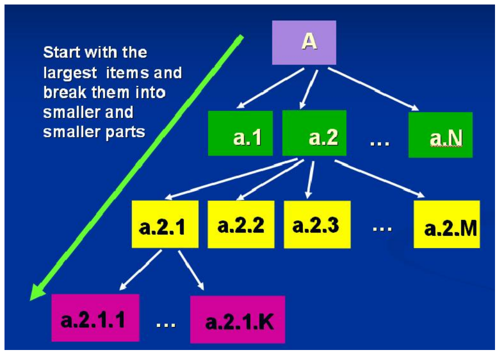
Top-Down Approach: Advantages and Issues

Advantages:

- Ensures projects are organized logically based on the nature of the project
- Promotes stakeholder participation in the planning phase of the project
- Can create a greater understanding of the entire project by all participants

Issues:

- Need to make sure major objectives are not forgotten
- Make sure to decompose the tasks to appropriate levels
- Can be time consuming, must guard against "analysis paralysis"
- Cost and time estimates are more difficult to create and generally less accurate than under the analogy approach


Start with the largest items and break them into smaller and smaller parts
A
a.1
a.2
…
a.N
a.2.1
a.2.2
a.2.3
…
a.2.M
a.2.1.1
…
a.2.1.K

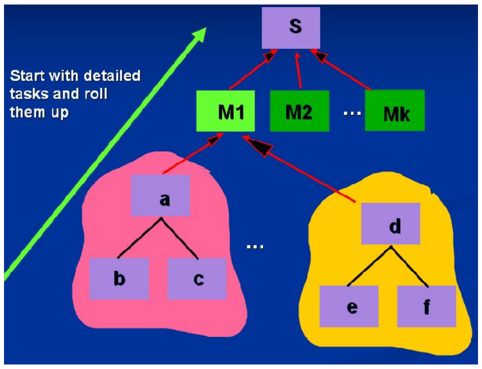
Bottom-Up Approach: Advantages and Issues

Advantages

- May lead to a more complete list of tasks and detailed description of tasks
- Promotes participation of various stakeholders in the planning phase of the project
- Can create a greater understanding of the entire project by all participants

Issues

- Difficult to retain focus on the "big picture"
- Need to make sure major objectives are not forgotten
- Harder to get organized into logical steps or phases



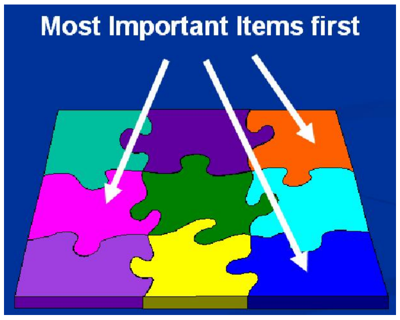
Thread Approach: Advantages and Issues

Advantages

- Generally the most important stakeholder objectives done first
- Greater control and focus of the brainstorming sessions
- Promotes stakeholder participation in the planning phase of the project
- Can create a greater understanding of the entire project by all participants

Issues

- Make sure to not lose focus of the "big picture"
- May lose site of the effect one objective may have on another
- Increases the need for communication
- More successful when the project leader and team has a good understanding of the project's objectives



Which Approach to Choose?

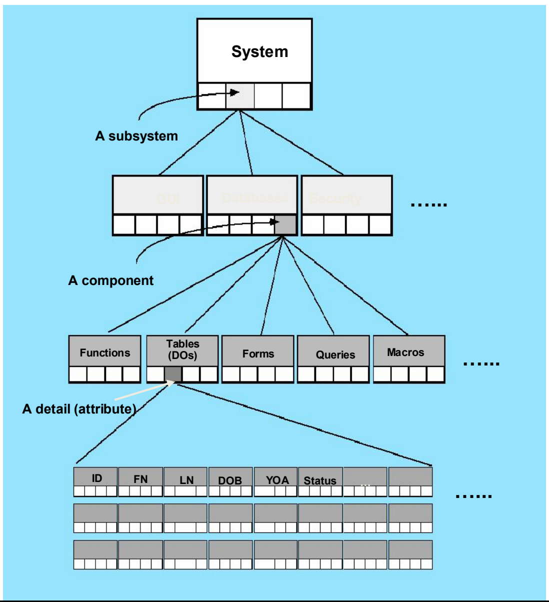
- **In case of the existence of a similar project:**
  would lead you to the analogy approach which if done correctly is the fastest and most accurate method

- **In case of an evolutionary type of project:**
  depends on experience level of the project manager and team:
  - if little experience, choose the top-down approach;
  - if many years of experience then choose a bottom-up approach

- **In case of a revolutionary type of project:**
  - if the product or process is very unique, never anything like it before in this company or by this team then choose the top-down approach

WBS Structure Example: Hierarchical Design-Based Form

Webster Software System -- A Hierarchical Design Model
(system, subsystems, and component design)



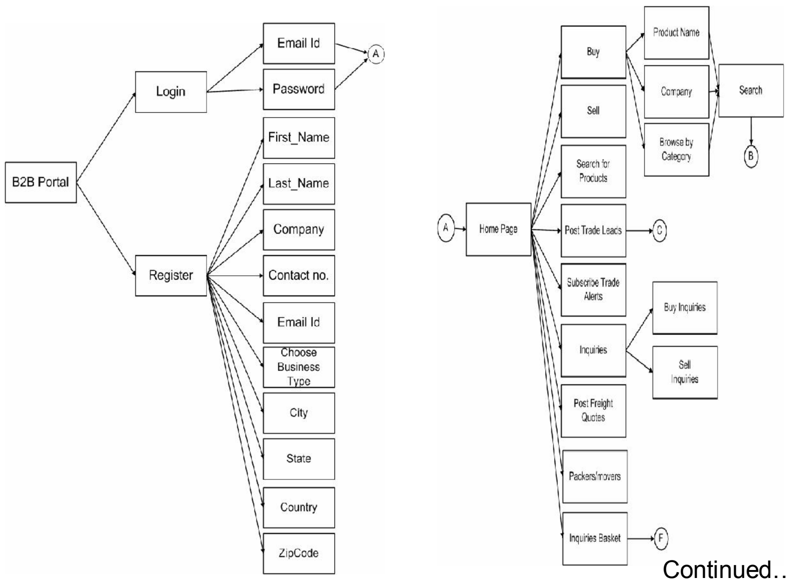
System Level
(Webster System)

Level of Subsystems (Domains)
(Databases, GUI, Security, HELP, etc.)

Level of Elements or Components
(tables, forms, queries, reports, macros and modules, …)

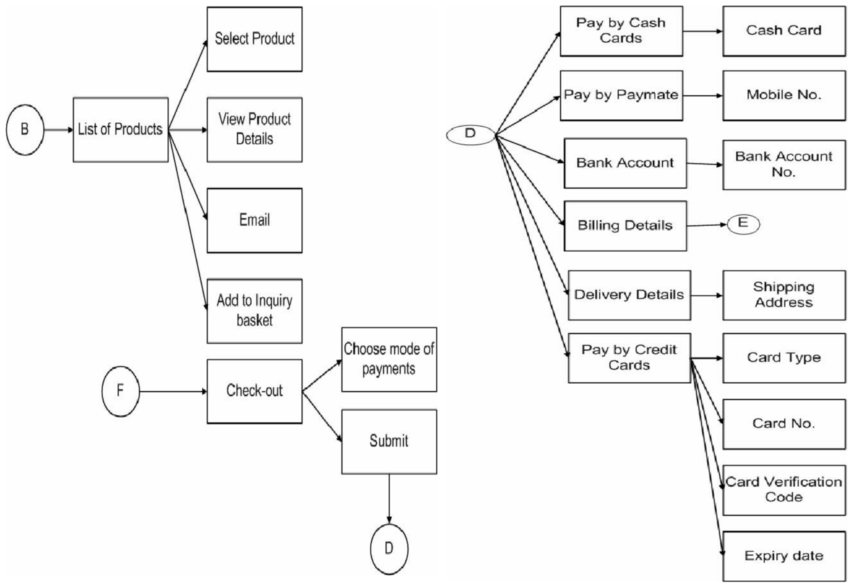
Level of Sub-elements, Details (for ex., attributes)
(ID, First Name, Last Name, DOB, YOA, status, …)

WBS Example: Decision Tree-Based Form

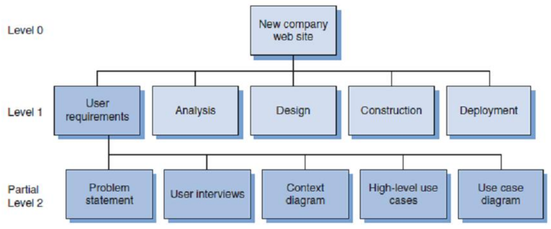


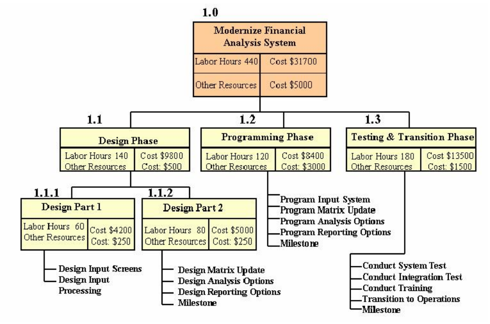
Continued….

WBS Example: GUI Hierarchical Design Model



WBS Examples: Process-Based Form





http://iteconcorp.com/T6WorkBreakdownStructure.html

WBS Example: Tabular Form

		Normal	Predecessors
1	**Project Initiation**		
1.1	Develop project charter	5 days	
1.2	Develop Statement Of Work	8days	1.1
1.3	Develop preliminary scope development	2days	1.2
1.4	Develop preliminary architectural model	5days	1.3
1.5	Project initiation complete	2days	1.4
2	**Project plan**		
2.1	Develop scope management plan	2days	1
2.2	Develop change management plan	5days	2.1
2.3	Develop intial descriptive budget	10days	2.1, 2.2
2.4	Develop schedule	3days	2.2, 2.3
2.5	Develop quality management plan	4days	2.4
2.6	Develop human resource plan	3days	2.5
2.7	Develop risk management plan	1 day	1
2.8	Project paln complete	1 day	2.1 - 2.7
3	**Project Execution**		
3.1	Release 1		
3.1.1	Analysis phase	15days	2
3.1.2	Design phase	10days	3.1.1
3.1.3	Construction phase	8days	3.1.2
3.1.4	Validation phase	15days	3.1.3
3.1.5	Deployment phase	3days	3.1.4
3.1.6	Closeout	1day	3.1.5
3.1.7	Release 1 Complete	2days	3.1.6
3.2	Release 2		
3.2.1	Analysis phase	15days	3
3.2.2	Design phase	10days	3.2.1
3.2.3	Construction phase	8days	3.2.2
3.2.4	Validation phase	15days	3.2.3
3.2.5	Deployment phase	3days	3.2.4
3.2.6	Closeout	1day	3.2.5
3.2.7	Release 2 Complete	2days	3.2.6
3.3	execution complete		
4	**project closeout**	2days	3
5	**project complete**	1days	4

Basic Principles for Creating a WBS

- The WBS represents 100% of the work required to produce the product.
  As soon as you define more than 100% of the scope, you have committed to doing more than you agreed to - scope creep has begun (100% Rule)
- Each WBS element represents a single deliverable
- Each deliverable is distinct
- Accountability for each task can be assigned to one team member
- Not all elements of the WBS need to be decomposed to the same depth
- Have all reporting and control mechanisms been included
- Be prepared for changes

Dictionary:
- Control accounts – accounting or finance department assigned account codes used in the accounting system to track costs
- Statement of work – describing the details of the work involved in creating each deliverable
- Responsible organization – who is responsible for each deliverable
- Schedule for major milestones
- Contract information if outside vendor involved
- Quality requirements
- Estimate of cost and resources required

Chapter 5.

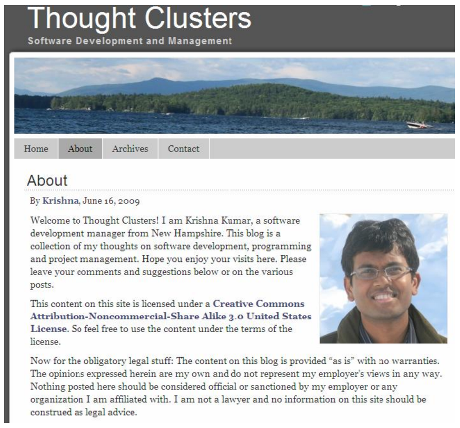
Project Planning.
Project Scope.
Additional (optional) information.

Project Management Plan: A Development Process Guidelines

- A project plan is a document used to **coordinate all project planning documents**
- Its main purpose is to ***guide project execution***
- Project plans assist the project manager in **leading the project team and assessing project status**
- Project performance should be measured against a baseline project plan
- Building the plan should ***not be done in secret or in isolation***; the whole project team needs to participate

SW Project Plan and SW Analysis Paralysis

http://www.thoughtclusters.com/2009/09/software-analysis-paralysis/



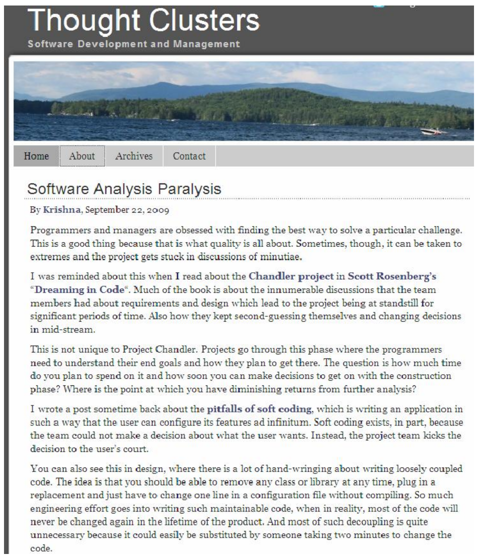
About

By **Krishna**, June 16, 2009

Welcome to Thought Clusters! I am Krishna Kumar, a software development manager from New Hampshire. This blog is a collection of my thoughts on software development, programming and project management. Hope you enjoy your visits here. Please leave your comments and suggestions below or on the various posts.

This content on this site is licensed under a **Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License**. So feel free to use the content under the terms of the license.

Now for the obligatory legal stuff: The content on this blog is provided "as is" with no warranties. The opinions expressed herein are my own and do not represent my employer's views in any way. Nothing posted here should be considered official or sanctioned by my employer or any organization I am affiliated with. I am not a lawyer and no information on this site should be construed as legal advice.



Software Analysis Paralysis

By **Krishna**, September 22, 2009

Programmers and managers are obsessed with finding the best way to solve a particular challenge. This is a good thing because that is what quality is all about. Sometimes, though, it can be taken to extremes and the project gets stuck in discussions of minutiae.

I was reminded about this when I read about the **Chandler project** in **Scott Rosenberg's** "**Dreaming in Code**". Much of the book is about the innumerable discussions that the team members had about requirements and design which lead to the project being at standstill for significant periods of time. Also how they kept second-guessing themselves and changing decisions in mid-stream.

This is not unique to Project Chandler. Projects go through this phase where the programmers need to understand their end goals and how they plan to get there. The question is how much time do you plan to spend on it and how soon you can make decisions to get on with the construction phase? Where is the point at which you have diminishing returns from further analysis?

I wrote a post sometime back about the **pitfalls of soft coding**, which is writing an application in such a way that the user can configure its features ad infinitum. Soft coding exists, in part, because the team could not make a decision about what the user wants. Instead, the project team kicks the decision to the user's court.

You can also see this in design, where there is a lot of hand-wringing about writing loosely coupled code. The idea is that you should be able to remove any class or library at any time, plug in a replacement and just have to change one line in a configuration file without compiling. So much engineering effort goes into writing such maintainable code, when in reality, most of the code will never be changed again in the lifetime of the product. And most of such decoupling is quite unnecessary because it could easily be substituted by someone taking two minutes to change the code.

What Is Project Scope?

- **Scope** - refers to ***all (100%)*** of the work involved in creating the products of the project and the processes used to create them
- A **deliverable** - is a particular product produced as part of a project, such as hardware or software, planning documents, or meeting minutes
- **Scope management plan** describes how the project team will
  - define the scope,
  - develop the detailed scope statement,
  - define and develop the work breakdown structure (WBS),
  - verify the scope, and
  - control the scope
- A **scope statement** describes the characteristics of the product that the project was created to deliver.

29

Scope Statement Depends on:

1. Project **size,** including the number of people, dollar value, duration, and geographic span
2. Degree of **risk** to the business
3. **Cash requirements**, such as length of time for return on investment and initial cash requirements
4. **Technology** utilized (for example, maturity, experience of current staff)
5. **Project team experience** with technology, business, and industry
6. **Nature of the deliverables**, such as whether this is a new product or service, an upgrade, or a repair
7. **Strategic importance** of the project to the organization
8. **Project definition** (for example, whether the requirements are undefined, partially defined, or poorly defined)

30