Team Homework (I)

- Term project: Instant Messenger
  - Survey IM and present your findings to decide which IM to choose for your term project (due 9/26)

- Progress report: Prepare meeting minutes for your team's company visit plan and term project plan (every week)
  - From the text book Chapter 5

- Read and present this paper: S.G. Eick ; T.L. Graves ; A.F. Karr ; J.S. Marron ; A. Mockus, Does code decay? Assessing the evidence from change management data, IEEE Transactions on Software Engineering, 1-12, 27(1), 2001 (due 10/3)

Team Homework (II)

- Compile an Excel file to compute your estimates with percentage confident estimation method (due 10/3)

- From the text book: 附錄A：軟體工程個案研究-需求管理
  - Read the case study and make a presentation (due 10/17)

- Prepare a WBS for your term project with estimates by using Percentage Confident estimation method, and potential risks for your term project. (due 10/17)

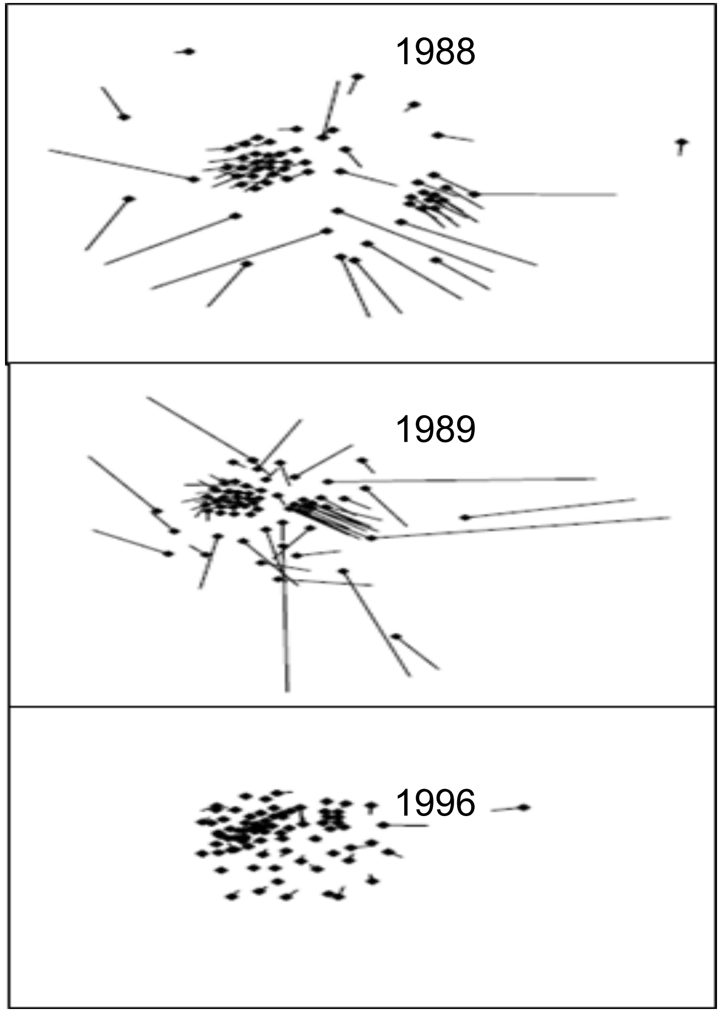
A Light-Weight Software Development Process

Prof. Jonathan Lee
CSIE Department
National Taiwan University

Motivation: Software/Code Decay

- A unit of code (in most cases, a module) is decayed if it is *harder to change than it should be*, measured in terms of effort, interval and quality.

- An Example: The Telephone Switches Project
  - Fifteen-year old real-time software system for telephone switches
  - 100,000,000 lines of source code (C/C++) and 100,000,000 lines of header and make files, organized into some 50 major subsystems and 5,000 modules.
  - More than 10,000 software developers have participated.
  - A module within one of the clusters is often changed together with other modules in the cluster but not with other modules
    - Head of each tadpole-like shape corresponds to a module



S.G. Eick ; T.L. Graves ; A.F. Karr ; J.S. Marron ; A. Mockus, **Does code decay? Assessing the evidence from change management data, IEEE Transactions on Software Engineering, 1-12, 27(1), 2001**

What is a Software Process?

- A *software process* is a set of activities, methods, practices and transformations that people employ to develop and maintain software and the associated products, including requirements documents, project plans, design documents, code, test cases, and etc.

- In Software Engineering, processes are the fundamental for almost everything, for example, software process, DevOps process, project management process, risk management process, change control process, and etc.

Software Development Process

- Engineering Practices
  - Requirements
  - System Architecture
  - Software Design
- Project Management
  - WBS
  - Meeting Minutes

Project Management

- Project Plan: Work Breakdown Structure (WBS)
  - Tasks breakdown
  - Estimation: effort, system size, schedule, cost
  - Responsibility: task assignment and commitment
  - Team Work

- Project Monitoring and Control: Meeting minutes
  - Objective
  - Agenda: from WBS or team members’ responsibility or risks ..
  - Issues: technical review, verification and validation, schedule deviation
  - Action Items: open, ongoing, suspended, close

Engineering Practices

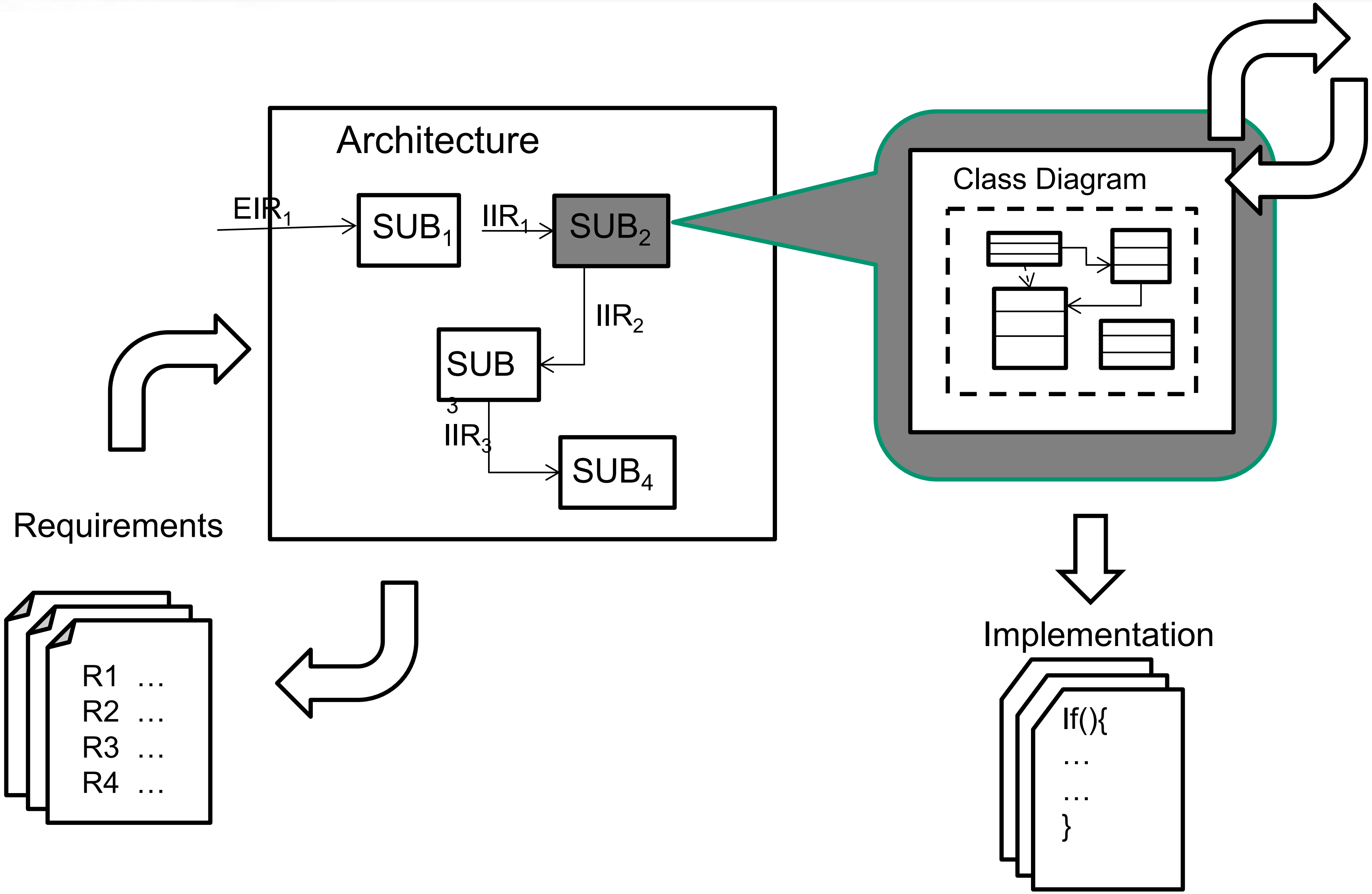
- Requirements:
  - How to elicit and refine requirements: never complete, always vague, and change all the time
  - How to document requirements: functional, nonfunctional, interface requirements
- System Architecture
  - Why do we need a system architecture?
  - How do we iterate between requirements and architecture?
- Software Diagram
  - Initial Design: based on requirements and system architecture
  - Redesign: Design issues and design principles

Engineering Practices Illustration

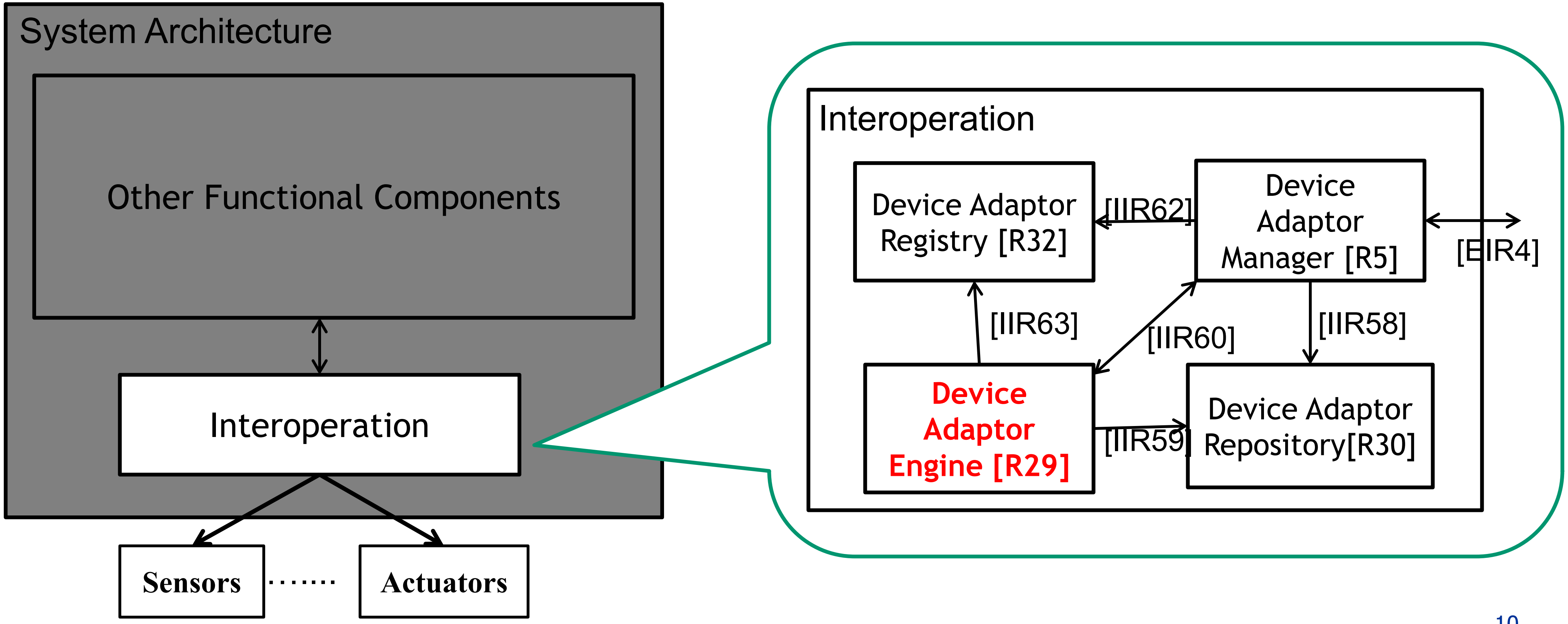
EIR: External Interface Requirement

IIR: Internal Interface Requirement

SUB: Sub-System



Refine System Architecture



From Requirements to Design: How to Iterate?

- Follow requirements to the letter to construct class diagrams in a stepwise way.
- Find out the most important operations and their scenario.
- Compile a sequence diagram of the scenario.
- List the design issues or concerns in conducting the design to serve as a basis for further discussion with requirements providers.

Where and How to Obtain Requirements?

- From our clients (customers), users, and general public

- How to obtain?
  - Traditional approach: Clients or customers pay us to develop software

  - More realistic and feasible approach: We, as software developers, pay our clients, customers, users, and general public to obtain requirements

Software Project Management

Prof. Jonathan Lee
CSIE Department
National Taiwan University

What is Project?

- Project – an endeavor undertaken to create a **unique** product or service
  - Has a definite beginning and end and interrelated activities
  - Cease when declared objectives have been attained
- Projects are unique – characteristics are **progressively elaborated**
- Scope of project should remain **constant**
  - How to determine the project scope? (WBS, System Architecture)

What is Project Management?

- Project Management: the application of knowledge, skills, tools and techniques to project activities in order to meet or exceed stakeholder's **needs and expectations** from a **defined project** – balancing the following:
  - **Scope, time, cost, and quality**
  - Requirements (needs) vs. unidentified requirements (expectations)
- A defined project: a project with a defined beginning and end, a defined scope and resources.

Software Projects Survey: Standish Group Chaos Report

- Software projects (in US): success rate means the project is completed **on time** and **within budget**.
- Projects being **challenged** means over budget and/or with less than the required features and functions.
- Projects being failed means cancelled prior to completion or delivered but never used.

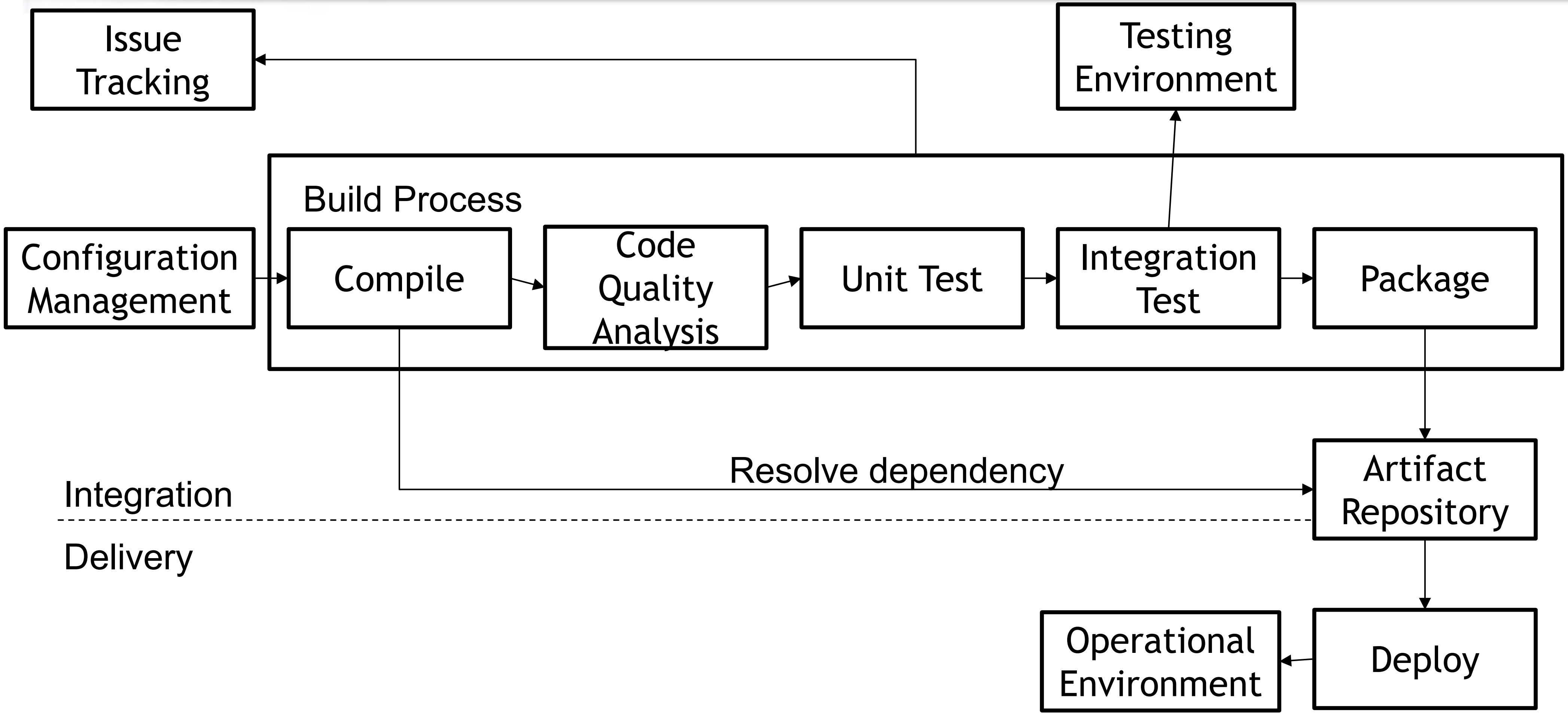
	Year 2015	Year 2014	Year 2010	Year 2008	Year 2002	Year 2000	Year 1998	Year 1996	Year 1994
Successful	29%	28%	37%	32%	34%	28%	26%	27%	16%
Challenged	52%	55%	42%	44%	15%	23%	28%	40%	31%
Failed	19%	17%	21%	24%	51%	49%	46%	33%	53%

Any Improvement?

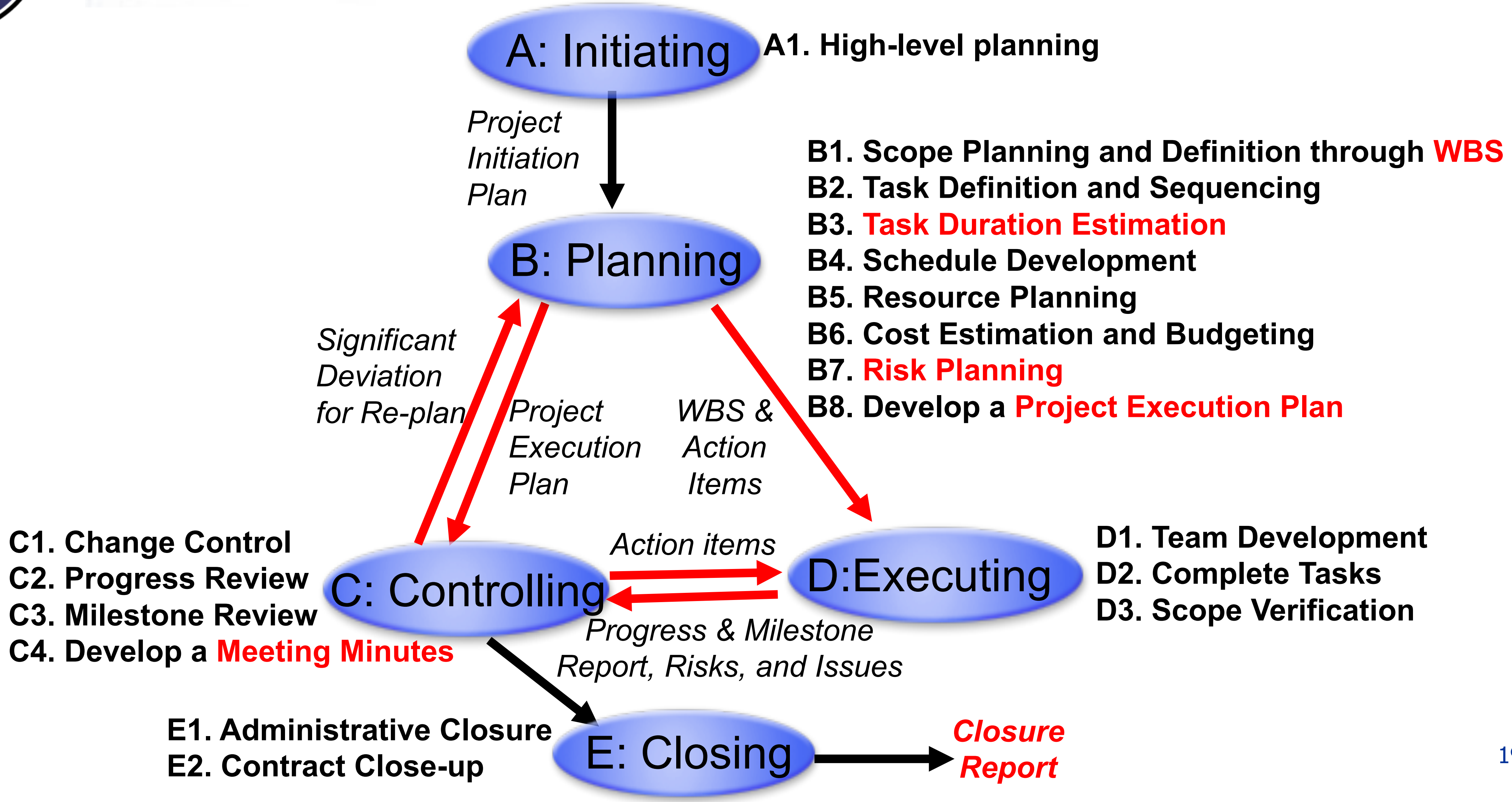
- What Software Engineering community has been doing so far to improve software project success rate.

- Engineering practice: Extreme programming, Aspect-oriented programming, Design patterns and frameworks, and etc.
- Project management: WBS, SCRUM
- Process improvement: Model-based process improvement (CMMI)
- Infrastructure: **DevOps** (continuous integration and delivery)

DevOps Process



Project Management Processes



Initiating Process

- High-Level Planning: Define the overall parameters of a project, for example,
  - Scope of the project: budget assigned, size of the software system, approximate schedule, and etc.
  - Have a Project Sponsor, and secure approval, ensure commitments.

- Deliverable of High-level planning is a **Project Initiation Plan** (or **Project Charter**).

Project Initiation Plan (Project Charter)

- A business case/need justifies the reason why to initiate a project.
- A Charter includes:
  - Business need/Business Case, Product description & title, Signed contract, Project Manager Identification & Authority level, Senior Management approval, Project's Goals and Objectives (goal, strategy, objective, tactic), Constraints, Assumptions.
- Ensure a consistent understanding of the project, set expectations, identify resources to move the project to the next level of detailed planning, and identify potential problems.

Planning Processes

- **Scope Planning and Definition through WBS**
- Task Definition
- **Project and Task Estimation**
- Schedule Development
- Resource Planning
- Cost Estimation and Budgeting
- **Risk Planning**
- **Develop a Project Execution Plan**

Why WBS is Important?

- Scope
  - Use WBS to establish a project's scope
- Estimation
  - Estimate required staffs, budget, and time based on WBS
- Scheduling
  - Develop project schedule based on WBS
- Teamwork
  - Assign tasks to team members based on WBS
- Project Monitoring and Control
  - Monitor project progress based on WBS (integrate with meeting minutes)
  - Control action items generated from WBS

What is WBS?

- Work Breakdown Structure (WBS) is a **deliverable-oriented** grouping of project assignments that organizes and defines the scope of the project
  - Each descending level represents further detail; smaller and more manageable pieces
  - Work products (deliverables) should also be explicitly described in the work package.

- WBS is a graphical picture of the **project hierarchy**
- WBS was first introduced by DoD in 1957.

How to Divide the Work?

- Usually, first Level is commonly the same as the processes as the **domain** involved in the project, for example, software lifecycle in developing a software system.
- Each level of a WBS is a smaller segment of the level above
- Break down project into **tasks (lowest-level element)** that
  - Are realistically and confidently **estimated**
  - Can be **completed** under 40/80 person-hour rule of thumb, that is, 5/10 person-day (one/two weeks), or can be varied based on the software process model adopted (e.g. Agile method may be less).
  - Have a meaningful conclusion and **deliverable**.
  - **Work assignments, tasks, and action items** usually refer to the same concept.

Work Package

- A task is described by a Work package, including
  - Task name
  - Description of work to be done
  - Preconditions for starting: Other Work packages that need to be completed before this task can be started
  - Duration
  - Required resources
  - Work product to be produced
  - Involved Risks
- Work package usually is the lowest level of WBS and corresponds to well defined work assignment (or task) for one worker for a week or two (80 person-hour).

Work Package Example

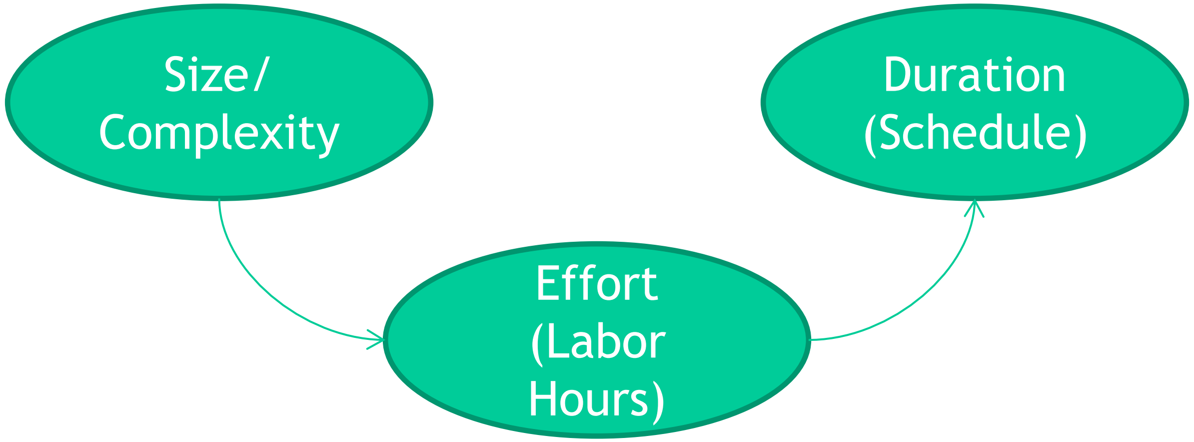
WBS #:	2.1	Task:	Develop Project Plan
Est. Level of Effort:	20 hrs	Owner:	Project Manager
Resources Needed:	Subject Matter Experts	Work Products:	Project Plan
Description of Task:	Develop a detailed project plan that lists all key resources, tasks, milestones, dependencies, and durations.		
Input:	• Approved Project Charter		
Dependencies:	• Approval of Budget		
Risk:	• IT Application implementation releases, which may be conflicting with implementation		

A WBS Example

- Meeting Scheduler
  - Survey
  - 專案規劃
    - 發展WBS
    - 發展專案計畫書
  - 需求分析
    - 需求擷取
    - 訂定目標
    - 定義角色
    - 發展使用案例
    - 發展Data Dictionary
    - 發展系統架構
    - 撰寫需求文件
    - M1：需求文件產出
  - 系統設計
  - 系統實作
  - 系統測試
  - M6：驗收測試

Project Estimation

- Estimate **size** or complexity first
- Estimate required **effort** based on the size estimation
- Estimate required **duration** based on the effort estimation



Planning Poker

- WHO: participants in planning poker include all developers on a team.
- HOW:
  - For each task, task owners answer any questions that the estimators have.
  - Each estimator write down his/her estimate on a card.
  - When every estimator has made a decision, all cards are simultaneously turned and shown.
  - The high and low estimators explain their estimates.
  - If the estimates do not converge, repeat the process until they do.
- One way to link to Structured Expert Judgment is by assigning the high and low to worst and best, and the average to most likely.

Structured Expert Judgment

- Single-Point Estimates
  - Problem: Tend to be best case estimates (25%)
- Best, Worst Case Estimates
  - Problem: Adding up best and worst cases would result in statistical anomalies (say, finish all 10 tasks, $(1/4)^{10}$)
- Best (b), Most Likely (m), Worst(w), Expected Case (e) Estimates
  - Expected Case: e = (b+4m+w)/6.
  - Problem: 6 σ (standard deviations) assuming that the estimation range would account for 99.7% of all possible outcomes (or 3 actual outcomes out of 1000 estimates).

Summary: Percentage Confident Estimates

- Small number of tasks (simplified version for < 10 tasks): only consider standard deviation using 6 sigma as the divisor for all tasks
- Large number of tasks (for >= 10 tasks, divisor for each task)
  - Compute e = (b+4m+w)/6 or e = (b+3m+2w)/6.
  - Compute σ (standard deviation) = (w – b)/ **Divisor**; variance v = σ^2; and compute aggregate σ by taking the squared root of the total v.
  - Compute **Divisor for each task** from the z score table based on percentage of your actual outcomes falling within your estimation range (e.g., 70% → z = 1.05 * 2 (2.1); 99.7% → z = 3 * 2 (6.0)).
  - Compute **Percentage Confident** (can be measured **ranging from 2% to 98%**) from the z score table (that is, e ± z σ).
    - 75% = e + 0.67 σ
    - 75 % = (50 + 25)%
    - 0.67 σ = 24.8% (from z score)

Z Score Table

z	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0	0z	0.004	0.008	0.012	0.016	0.0199	0.0239	0.0279	0.0319	0.0359
0.1	0.0398	0.0438	0.0478	0.0517	0.0557	0.0596	0.0636	0.0675	0.0714	0.0753
0.2	0.0793	0.0832	0.0871	0.091	0.0948	0.0987	0.1026	0.1064	0.1103	0.1141
0.3	0.1179	0.1217	0.1255	0.1293	0.1331	0.1368	0.1406	0.1443	0.148	0.1517
0.4	0.1554	0.1591	0.1628	0.1664	0.17	0.1736	0.1772	0.1808	0.1844	0.1879
0.5	0.1915	0.195	0.1985	0.2019	0.2054	0.2088	0.2123	0.2157	0.219	0.2224
0.6	0.2257	0.2291	0.2324	0.2357	0.2389	0.2422	0.2454	0.2486	0.2517	0.2549
0.7	0.258	0.2611	0.2642	0.2673	0.2704	0.2734	0.2764	0.2794	0.2823	0.2852
0.8	0.2881	0.291	0.2939	0.2967	0.2995	0.3023	0.3051	0.3078	0.3106	0.3133
0.9	0.3159	0.3186	0.3212	0.3238	0.3264	0.3289	0.3315	0.334	0.3365	0.3389
1	0.3413	0.3438	0.3461	0.3485	0.3508	0.3531	0.3554	0.3577	0.3599	0.3621
1.1	0.3643	0.3665	0.3686	0.3708	0.3729	0.3749	0.377	0.379	0.381	0.383
1.2	0.3849	0.3869	0.3888	0.3907	0.3925	0.3944	0.3962	0.398	0.3997	0.4015
1.3	0.4032	0.4049	0.4066	0.4082	0.4099	0.4115	0.4131	0.4147	0.4162	0.4177
1.4	0.4192	0.4207	0.4222	0.4236	0.4251	0.4265	0.4279	0.4292	0.4306	0.4319
1.5	0.4332	0.4345	0.4357	0.437	0.4382	0.4394	0.4406	0.4418	0.4429	0.4441
1.6	0.4452	0.4463	0.4474	0.4484	0.4495	0.4505	0.4515	0.4525	0.4535	0.4545
1.7	0.4554	0.4564	0.4573	0.4582	0.4591	0.4599	0.4608	0.4616	0.4625	0.4633
1.8	0.4641	0.4649	0.4656	0.4664	0.4671	0.4678	0.4686	0.4693	0.4699	0.4706
1.9	0.4713	0.4719	0.4726	0.4732	0.4738	0.4744	0.475	0.4756	0.4761	0.4767
2	0.4772	0.4778	0.4783	0.4788	0.4793	0.4798	0.4803	0.4808	0.4812	0.4817
2.1	0.4821	0.4826	0.483	0.4834	0.4838	0.4842	0.4846	0.485	0.4854	0.4857
2.2	0.4861	0.4864	0.4868	0.4871	0.4875	0.4878	0.4881	0.4884	0.4887	0.489
2.3	0.4893	0.4896	0.4898	0.4901	0.4904	0.4906	0.4909	0.4911	0.4913	0.4916
2.4	0.4918	0.492	0.4922	0.4925	0.4927	0.4929	0.4931	0.4932	0.4934	0.4936
2.5	0.4938	0.494	0.4941	0.4943	0.4945	0.4946	0.4948	0.4949	0.4951	0.4952
2.6	0.4953	0.4955	0.4956	0.4957	0.4959	0.496	0.4961	0.4962	0.4963	0.4964
2.7	0.4965	0.4966	0.4967	0.4968	0.4969	0.497	0.4971	0.4972	0.4973	0.4974
2.8	0.4974	0.4975	0.4976	0.4977	0.4977	0.4978	0.4979	0.4979	0.498	0.4981
2.9	0.4981	0.4982	0.4982	0.4983	0.4984	0.4984	0.4985	0.4985	0.4986	0.4986
3	0.4987	0.4987	0.4987	0.4988	0.4988	0.4989	0.4989	0.4989	0.499	0.499
3.1	0.499	0.4991	0.4991	0.4991	0.4992	0.4992	0.4992	0.4992	0.4993	0.4993
3.2	0.4993	0.4993	0.4994	0.4994	0.4994	0.4994	0.4994	0.4995	0.4995	0.4995
3.3	0.4995	0.4995	0.4995	0.4996	0.4996	0.4996	0.4996	0.4996	0.4996	0.4997
3.4	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4998

Single Point Estimates

- Down to the task-level estimates and have the people who actually do the work create the estimates.
- Single-point estimates tend to be akin to Best Case estimates (25% likely).

Example of Single-Point Estimates

Task	Estimated Days to Complete
Task 1	1.5
Task 2	1.5
Task 3	2.0
Task 4	0.75
Task 5	0.5
Task 6	0.25
Task 7	2.0
Task 8	1.0
Task 9	0.75
Task 10	1.25
TOTAL	11.25

Best, Worst Case Estimates

- Best Case estimates is 25% likely, that is, have only a 25% chance of doing as well or better than the estimates. Worst Case estimates: How long the task would take if everything goes wrong.
- Prepare for the worst (75%), hope for the best (25%).
  - Going through the process of writing down best case and worst case estimates helps develop the habit of thinking through the worst case outcome when estimating.
- Problem of adding up best case and worst case estimates.
  - The odds to deliver any individual task according to a best case estimate is 25% (1 in 4).
  - What is the odds to deliver 10 tasks based on their best case estimates? $(1/4)^{10}$.

Example of Best Case and Worst Case

Task	Estimated Days to Complete	
	Best Case	Worst Case
Task 1	1.25	2.0
Task 2	1.5	2.5
Task 3	2.0	3.0
Task 4	0.75	2.0
Task 5	0.5	1.25
Task 6	0.25	0.5
Task 7	1.5	2.5
Task 8	1.0	1.5
Task 9	0.5	1.0
Task 10	1.25	2.0
TOTAL	10.5	18.25

Best, Most Likely, Worst, Expected Case Estimates

- $e = (b + 4 * m + w) / 6$
  - e: ExpectedCase; b: BestCase; m: MostLikelyCase; w: WorstCase.
  - Account for the full width of the range (that is, from the best case to worst case estimates) and the position of the Most Likely Case within the range.
- People’s most likely case estimates tend to be optimistic.
- Another way to compute expected case estimates.
  - $e = (b + 3 * m + 2 * w) / 6$, that is, put more weight on the worst case and less weight on most-likely case.

Example of Best Case, Worst Case, and Most Likely Case

(0.75 + 4 * 1 + 2.0) / 6 = 1.13

Estimated Days to Complete				
Task	Best Case	Most Likely Case	Worst Case	Expected Case
Task 1	1.25	1.5	2.0	1.54
Task 2	1.5	1.75	2.5	1.83
Task 3	2.0	2.25	3.0	2.33
Task 4	0.75	1	2.0	1.13
Task 5	0.5	0.75	1.25	0.79
Task 6	0.25	0.5	0.5	0.46
Task 7	1.5	2	2.5	2.00
Task 8	1.0	1.25	1.5	1.25
Task 9	0.5	0.75	1.0	0.75
Task 10	1.25	1.5	2.0	1.54
TOTAL	10.5	13.25	18.25	13.62

MRE: Compare Estimates to Actuals

- MRE (Magnitude of Relative Error) = Absolute[(ActualResult – ExpectedCase)/ActualResult]
- MRE helps set up **a feedback loop** to improve the estimates over time.
  - 29% on next page can be used to measure the **accuracy** of the estimates.
- The right-most column shows how many estimates are **in the best case/worst case range**, that is, 8 out of 10 tasks estimation, or 80% confident.
- The feedback should be as timely as possible to be effective.

Example of Tracking Accuracy of Individual Estimates

	Estimated Days to Complete					
Task	Best Case	Worst Case	Expected Case	Actual Outcome	MRE	In Range from Best Case to Worst Case?
Task 1	1.25	2	1.54	2	23%	Yes
Task 2	1.5	2.5	1.83	2.5	27%	Yes
Task 3	2	3	2.33	1.25	87%	No
Task 4	0.75	2	1.13	1.5	25%	Yes
Task 5	0.5	1.25	0.79	1	21%	Yes
Task 6	0.25	0.5	0.46	0.5	8%	Yes
Task 7	1.5	2.5	2.00	3	33%	No
Task 8	1	1.5	1.25	1.5	17%	Yes
Task 9	0.5	1	0.75	1	25%	Yes
Task 10	1.25	2	1.54	2	23%	Yes
TOTAL	10.50	18.25	13.625	16.25		80%
Average Accuracy					29%	

Decomposition and Recomposition

- **Decomposition** is the practice of separating an estimate into multiple pieces, estimating each piece individually, and then **recombining** the individual estimates into an aggregate estimate, that is, the tasks that are broken down in the WBS.
- The **Law of Large Numbers** means that if you create one big estimate, the estimates error tendency will be completely on the high side or completely on the low side.
- But, it you create several smaller estimates, some of the estimation errors will be on the high side, and some will be on the low side. The errors will tend to cancel each other out to some degree.

Refinement

- Software projects tend to **progress** from large-grain focus at the beginning to fine-grain focus at the end.
- **Software development is a process of making larger numbers of steadily smaller decisions.**
- A process of steady refinement is that the further into the project you are, the finer grained your decomposed estimates can be.
- Need 5 to 10 individuals items in order to benefit from the Law of Large Numbers.

Example of Estimation by Decomposition Using Best Case, Expected Case, and Worst Case Estimates

Weeks to Complete				
Task	Best Case (25% Likely)	Most Likely Case	Worst Case (75% Likely)	Expected Case (50% Likely)
Task 1	1.6	2.0	3.0	2.10
Task 2	1.8	2.5	4.0	2.63
Task 3	2.0	3.0	4.2	3.03
Task 4	0.8	1.2	1.6	1.20
Task 5	3.8	4.5	5.2	4.50
Task 6	3.8	5.0	6.0	4.97
Task 7	2.2	2.4	3.4	2.53
Task 8	0.8	1.2	2.2	1.30
Task 9	1.6	2.5	3.0	2.43
Task 10	1.6	4.0	6.0	3.93
TOTAL	20.0	28.3	38.6	28.62

Percentage Confident Estimates

- Small numbers of tasks (<10 tasks)
  - Standard deviation = (ΣWorstCaseEstimate – ΣBestCaseEstimate) / 6.
  - Use **6 as the divisor**
- Large number of tasks (≥ 10 tasks)
  - IndividualStandardDeviation = (IndividualWorstCaseEstimate – IndividualBestCaseEstimate) / 6.
  - For each individual task, use 6 as the divisor
- Large number of tasks (≥ 10 tasks), and with confident percentage from previous projects
  - For each individual task, use a divisor that is based on our previous projects confident percentage.

Small Number of Tasks

- Standard deviation = (ΣWorstCaseEstimate – ΣBestCaseEstimate) / 6.
  - From the table on page 44, we have (38.6 – 20.0)/6 = 3.1.
- We can use a table of standard deviations to compute a percentage likelihood.
  - For example (page 44), a statistically valid **75%** likely estimate would be:

  Expected case + 0.67 * StandardDeviation, which is 28.62 + (0.67 * 3.1) = 30.7weeks.

  - Refer to Z score Table for 0.67 on pages 47, 48 and 49.

A Table of Standard Deviations

Percentage Confident	Calculation
2%	Expected case - (2 x StandardDeviation)
10%	Expected case - (1.28 x StandardDeviation)
16%	Expected case - (1 x StandardDeviation)
20%	Expected case - (0.84 x StandardDeviation)
25%	Expected case - (0.67 x StandardDeviation)
30%	Expected case - (0.52 x StandardDeviation)
40%	Expected case - (0.25 x StandardDeviation)
50%	Expected case - 0
60% = 50% + 10% (0.098)	Expected case + (0.25 x StandardDeviation)
70% = 50% + 20% (0.1985)	Expected case + (0.52 x StandardDeviation)
75% = 50% + 25% (0.2486)	Expected case + (*0.67* x StandardDeviation)
80%	Expected case + (0.84 x StandardDeviation)
84%	Expected case + (1 x StandardDeviation)
90%	Expected case + (1.28 x StandardDeviation)
98%	Expected case + (2 x StandardDeviation)

Z Score Table

z	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0	0z	0.004	0.008	0.012	0.016	0.0199	0.0239	0.0279	0.0319	0.0359
0.1	0.0398	0.0438	0.0478	0.0517	0.0557	0.0596	0.0636	0.0675	0.0714	0.0753
0.2	0.0793	0.0832	0.0871	0.091	0.0948	0.0987	0.1026	0.1064	0.1103	0.1141
0.3	0.1179	0.1217	0.1255	0.1293	0.1331	0.1368	0.1406	0.1443	0.148	0.1517
0.4	0.1554	0.1591	0.1628	0.1664	0.17	0.1736	0.1772	0.1808	0.1844	0.1879
0.5	0.1915	0.195	0.1985	0.2019	0.2054	0.2088	0.2123	0.2157	0.219	0.2224
0.6	0.2257	0.2291	0.2324	0.2357	0.2389	0.2422	0.2454	0.2486	0.2517	0.2549
0.7	0.258	0.2611	0.2642	0.2673	0.2704	0.2734	0.2764	0.2794	0.2823	0.2852
0.8	0.2881	0.291	0.2939	0.2967	0.2995	0.3023	0.3051	0.3078	0.3106	0.3133
0.9	0.3159	0.3186	0.3212	0.3238	0.3264	0.3289	0.3315	0.334	0.3365	0.3389
1	0.3413	0.3438	0.3461	0.3485	0.3508	0.3531	0.3554	0.3577	0.3599	0.3621
1.1	0.3643	0.3665	0.3686	0.3708	0.3729	0.3749	0.377	0.379	0.381	0.383
1.2	0.3849	0.3869	0.3888	0.3907	0.3925	0.3944	0.3962	0.398	0.3997	0.4015
1.3	0.4032	0.4049	0.4066	0.4082	0.4099	0.4115	0.4131	0.4147	0.4162	0.4177
1.4	0.4192	0.4207	0.4222	0.4236	0.4251	0.4265	0.4279	0.4292	0.4306	0.4319
1.5	0.4332	0.4345	0.4357	0.437	0.4382	0.4394	0.4406	0.4418	0.4429	0.4441
1.6	0.4452	0.4463	0.4474	0.4484	0.4495	0.4505	0.4515	0.4525	0.4535	0.4545
1.7	0.4554	0.4564	0.4573	0.4582	0.4591	0.4599	0.4608	0.4616	0.4625	0.4633
1.8	0.4641	0.4649	0.4656	0.4664	0.4671	0.4678	0.4686	0.4693	0.4699	0.4706
1.9	0.4713	0.4719	0.4726	0.4732	0.4738	0.4744	0.475	0.4756	0.4761	0.4767
2	0.4772	0.4778	0.4783	0.4788	0.4793	0.4798	0.4803	0.4808	0.4812	0.4817
2.1	0.4821	0.4826	0.483	0.4834	0.4838	0.4842	0.4846	0.485	0.4854	0.4857
2.2	0.4861	0.4864	0.4868	0.4871	0.4875	0.4878	0.4881	0.4884	0.4887	0.489
2.3	0.4893	0.4896	0.4898	0.4901	0.4904	0.4906	0.4909	0.4911	0.4913	0.4916
2.4	0.4918	0.492	0.4922	0.4925	0.4927	0.4929	0.4931	0.4932	0.4934	0.4936
2.5	0.4938	0.494	0.4941	0.4943	0.4945	0.4946	0.4948	0.4949	0.4951	0.4952
2.6	0.4953	0.4955	0.4956	0.4957	0.4959	0.496	0.4961	0.4962	0.4963	0.4964
2.7	0.4965	0.4966	0.4967	0.4968	0.4969	0.497	0.4971	0.4972	0.4973	0.4974
2.8	0.4974	0.4975	0.4976	0.4977	0.4977	0.4978	0.4979	0.4979	0.498	0.4981
2.9	0.4981	0.4982	0.4982	0.4983	0.4984	0.4984	0.4985	0.4985	0.4986	0.4986
3	0.4987	0.4987	0.4987	0.4988	0.4988	0.4989	0.4989	0.4989	0.499	0.499
3.1	0.499	0.4991	0.4991	0.4991	0.4992	0.4992	0.4992	0.4992	0.4993	0.4993
3.2	0.4993	0.4993	0.4994	0.4994	0.4994	0.4994	0.4994	0.4995	0.4995	0.4995
3.3	0.4995	0.4995	0.4995	0.4996	0.4996	0.4996	0.4996	0.4996	0.4996	0.4997
3.4	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4998

Large Number of Tasks: Decomposition and Recomposition

- IndividualStandardDeviation = (IndividualWorstCaseEstimate – IndividualBestCaseEstimate) / **6**.
- Notice that we use 6 as the divisor.
- To obtain the aggregate standard deviation, we follow the steps below:
  - (1) The standard deviation of each individual task using the above formula (σ)
  - (2) The square of each task's standard deviation, that is, the variance, v, ($\sigma^2 = v$)
  - (3) Total the variance, Σ v
  - (4) The aggregate σ is squared root of Σ v

Example of Complex Standard Deviation Calculations

(1.6 – 0.8) /6 =
0.133

Weeks to Complete				
Feature	Best Case	Worst Case	Standard Deviation	Variance (Standard Deviation Squared)
Task 1	1.6	3.0	0.233	0.054
Task 2	1.8	4.0	0.367	0.134
Task 3	2.0	4.2	0.367	0.134
Task 4	0.8	1.6	0.133 (1)	0.018 (2)
Task 5	3.8	5.2	0.233	0.054
Task 6	3.8	6.0	0.367	0.134
Task 7	2.2	3.4	0.200	0.040
Task 8	0.8	2.2	0.233	0.054
Task 9	1.6	3.0	0.233	0.054
Task 10	1.6	6.0	0.733	0.538
TOTAL	20.0	38.6	-	1.22 (3)
Standard Deviation	-	-	-	1.1 (4)

Problem with Using 6 Sigma as Divisor

- Statistically speaking, by using the 6 sigma(σ) formula for each task we are assuring that the person who created the best case and worst case estimates include a **6 standard deviation range** from best case to worst case.
- In this case, the estimation range would have to account for 99.7% of all possible outcomes, that is, 3 out of 1000 could fall outside their estimated ranges. (Refer to z score, number of standard deviation from the mean)
- However, in this case, assuming we have 2 out of 10 estimates, that is, 80% percentage confident.
- **Which divisor should we use in order to come up with a more feasible estimate?**
  - We need to re-calculate the standard deviation based on the 80% percentage confident.

Percentage Confident Estimates

- Based on the Percentage Confident to decide the divisor.
- For example, 70% → 2.1 (1.05*2); 80% → 2.6 (1.29*2); 90% → 3.3 (1.65*2); and 99.7% → 6.0 (3*2)(z = 3 standard deviation).
- Therefore, we will obtain the new aggregate standard deviation for an 80% confident 2.55. (page 54)
- For any statistically valid "percentage" likely estimate would be (page 48):
  - 75%: 28.6 + (0.67 * 2.55) = 30.3 weeks.
  - 80%: 28.6 + (0.84 * 2.55) = 30.7 weeks.
  - 98%: 28.6 + (2 * 2.55) = 33.7 weeks.

Z Score Table

z	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0	0z	0.004	0.008	0.012	0.016	0.0199	0.0239	0.0279	0.0319	0.0359
0.1	0.0398	0.0438	0.0478	0.0517	0.0557	0.0596	0.0636	0.0675	0.0714	0.0753
0.2	0.0793	0.0832	0.0871	0.091	0.0948	0.0987	0.1026	0.1064	0.1103	0.1141
0.3	0.1179	0.1217	0.1255	0.1293	0.1331	0.1368	0.1406	0.1443	0.148	0.1517
0.4	0.1554	0.1591	0.1628	0.1664	0.17	0.1736	0.1772	0.1808	0.1844	0.1879
0.5	0.1915	0.195	0.1985	0.2019	0.2054	0.2088	0.2123	0.2157	0.219	0.2224
0.6	0.2257	0.2291	0.2324	0.2357	0.2389	0.2422	0.2454	0.2486	0.2517	0.2549
0.7	0.258	0.2611	0.2642	0.2673	0.2704	0.2734	0.2764	0.2794	0.2823	0.2852
0.8	0.2881	0.291	0.2939	0.2967	0.2995	0.3023	0.3051	0.3078	0.3106	0.3133
0.9	0.3159	0.3186	0.3212	0.3238	0.3264	0.3289	0.3315	0.334	0.3365	0.3389
1	0.3413	0.3438	0.3461	0.3485	0.3508	0.3531	0.3554	0.3577	0.3599	0.3621
1.1	0.3643	0.3665	0.3686	0.3708	0.3729	0.3749	0.377	0.379	0.381	0.383
1.2	0.3849	0.3869	0.3888	0.3907	0.3925	0.3944	0.3962	0.398	0.3997	0.4015
1.3	0.4032	0.4049	0.4066	0.4082	0.4099	0.4115	0.4131	0.4147	0.4162	0.4177
1.4	0.4192	0.4207	0.4222	0.4236	0.4251	0.4265	0.4279	0.4292	0.4306	0.4319
1.5	0.4332	0.4345	0.4357	0.437	0.4382	0.4394	0.4406	0.4418	0.4429	0.4441
1.6	0.4452	0.4463	0.4474	0.4484	0.4495	0.4505	0.4515	0.4525	0.4535	0.4545
1.7	0.4554	0.4564	0.4573	0.4582	0.4591	0.4599	0.4608	0.4616	0.4625	0.4633
1.8	0.4641	0.4649	0.4656	0.4664	0.4671	0.4678	0.4686	0.4693	0.4699	0.4706
1.9	0.4713	0.4719	0.4726	0.4732	0.4738	0.4744	0.475	0.4756	0.4761	0.4767
2	0.4772	0.4778	0.4783	0.4788	0.4793	0.4798	0.4803	0.4808	0.4812	0.4817
2.1	0.4821	0.4826	0.483	0.4834	0.4838	0.4842	0.4846	0.485	0.4854	0.4857
2.2	0.4861	0.4864	0.4868	0.4871	0.4875	0.4878	0.4881	0.4884	0.4887	0.489
2.3	0.4893	0.4896	0.4898	0.4901	0.4904	0.4906	0.4909	0.4911	0.4913	0.4916
2.4	0.4918	0.492	0.4922	0.4925	0.4927	0.4929	0.4931	0.4932	0.4934	0.4936
2.5	0.4938	0.494	0.4941	0.4943	0.4945	0.4946	0.4948	0.4949	0.4951	0.4952
2.6	0.4953	0.4955	0.4956	0.4957	0.4959	0.496	0.4961	0.4962	0.4963	0.4964
2.7	0.4965	0.4966	0.4967	0.4968	0.4969	0.497	0.4971	0.4972	0.4973	0.4974
2.8	0.4974	0.4975	0.4976	0.4977	0.4977	0.4978	0.4979	0.4979	0.498	0.4981
2.9	0.4981	0.4982	0.4982	0.4983	0.4984	0.4984	0.4985	0.4985	0.4986	0.4986
3	0.4987	0.4987	0.4987	0.4988	0.4988	0.4989	0.4989	0.4989	0.499	0.499
3.1	0.499	0.4991	0.4991	0.4991	0.4992	0.4992	0.4992	0.4992	0.4993	0.4993
3.2	0.4993	0.4993	0.4994	0.4994	0.4994	0.4994	0.4994	0.4995	0.4995	0.4995
3.3	0.4995	0.4995	0.4995	0.4996	0.4996	0.4996	0.4996	0.4996	0.4996	0.4997
3.4	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4997	0.4998

Example of Computing Standard Deviation Using a Divisor

(1.6 – 0.8) / **2.6** = 0.308

Weeks to Complete				
Task	Best Case	Worst Case	Standard Deviation	Variance (Standard Deviation Squared)
Task 1	1.6	3.0	0.538	0.290
Task 2	1.8	4.0	0.846	0.716
Task 3	2.0	4.2	0.846	0.716
Task 4	0.8	1.6	0.308	0.095
Task 5	3.8	5.2	0.538	0.290
Task 6	3.8	6.0	0.846	0.716
Task 7	2.2	3.4	0.462	0.213
Task 8	0.8	2.2	0.538	0.290
Task 9	1.6	3.0	0.538	0.290
Task 10	1.6	6.0	1.692	2.864
TOTAL	20.0	38.6	-	6.48
Standard Deviation	-	-	-	2.55

Example of Percentage-Confident Estimates

Percentage Confident	Effort Estimate (weeks)
2%	23.5
10%	25.4
16%	26.1
20%	26.5
25%	26.9
30%	27.3
40%	28.0
50%	28.6
60%	29.3
70%	30.0
75%	30.3
80%	30.7
84%	31.2
90%	31.8
98%	33.7

WBS with Estimation

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	TransOn	Item	Task	Best	Worst	Most l	Expect	STD	VAR	Person ir	Start Date	Complete%	Actual
2		Survey											
3			Existing project comparison	1	2	1.5	1.8	0.4348	1.7000	Ray	10/13	100%	1
4			DSP Lib										
5			Android Speech Recognizer	1	3	1	2.3	0.8696	0.7561	Ian	10/13	100%	3
6			CMU Sphinx	1	2	2	1.8	0.4348	0.1890	Hsuan-Cl	10/13	100%	2
7			Others	1.5	9	3	6.8	3.2609	10.6333	Ray	10/13	100%	3
8		Learning											
9			Android programming	2	4	3	3.5	0.8696	0.7561	Uiling	10/13	100%	2
10		Requirements Analysis											
11			Extract Requirement	1	3	1	2.3	0.8696	0.7561	Carson	10/27	100%	1
12			Set goal	1	2	1.5	1.8	0.4348	0.1890	Uiling	10/13	100%	1
13			Define character	1	2	1.5	1.8	0.4348	0.1890	Uiling	10/13	100%	2
14			Usecase										
15			Write creator use case	1.5	3	3	2.8	0.6522	0.4253	Chih-Hsie	10/13	100%	3
16			Write participant use case	1.5	3	3	2.8	0.6522	0.4253	Chih-Hsie	10/13	100%	3
17			Write talking use case	1.5	3	3	2.8	0.6522	0.4253	Chih-Hsie	10/13	100%	2
18			Write receiving use case	1.5	3	3	2.8	0.6522	0.4253	Chih-Hsie	10/13	100%	3
19			Write requirement document	1	3	2	2.5	0.8696	0.7561	Carson	10/13	100%	2
20			M1:finish requirement document										
21		System Architecture											
22			Server part architecture	1	3	2	2.5	0.8696	0.7561	Mars	10/13	100%	1
23			Client part architecture	2	4	3	3.5	0.8696	0.7561	Sandy	10/13	100%	4
24			M2:finish requirement document								12/19		
25		ClassDiagram											
26			Client										
27			Login	1	4	3	3.3	1.3043	1.7013	Ian	10/13	100%	3
28			View	2	3	3	2.8	0.4348	0.1890	Sandy	10/13	100%	3
29			Meeting	4	8	6	7.0	1.7391	3.0246	Chih-Hsie	10/13	100%	6
30			History	3	5	4	4.5	0.8696	0.7561	Carson	10/13	100%	3
31			Participant	4	6	5	5.5	0.8696	0.7561	Mars	10/13	100%	4
32			MeetingRoom	2	3	3	2.8	0.4348	0.1890	HsuanCh	10/13	100%	3
33			Digital Speech Processing	3	4	5	4.0	0.4348	0.1890	Ian	10/13	100%	4
34			Connection	4	5	6	5.0	0.4348	0.1890	Uiling	10/13	100%	5
35			Server										
41													
42		Implementation											
43			Client										
72			Server										
73			DataBase	2	4	3	3.5	0.8696	0.7561	Chih-Hsie	12/29	100%	4
74			DataBean	2	4	4	3.7	0.8696	0.7561	Chih-Hsie	12/25	100%	4
75			History	2	4	4	3.7	0.8696	0.7561	Hsuan-Cl	12/26	100%	2
76			Meeting Room	2	4	3	3.5	0.8696	0.7561	Hsuan-Cl	12/29	100%	3
77			Connector	3	5	4	4.5	0.8696	0.7561	Ray	12/26	100%	4
78			Packet	3	5	4	4.5	0.8696	0.7561	Ray	12/24	100%	5

O	P
Total VAR	39.4127
Total Expected Effort	164.8
Percentage Confident	75%
Percentage Confident Effort	168.96
Divisor	2.3
Z(0.375)	1.15

Percentage Confidnet	Effort Estimate(man days)	
2%	151.88	-2.05
10%	156.71	-1.28
16%	158.53	-0.99
20%	159.48	-0.84
25%	160.54	-0.67
30%	161.49	-0.52
40%	163.18	-0.25
50%	164.75	0
60%	166.32	0.25
70%	168.01	0.52
75%	168.96	0.67
80%	170.02	0.84
84%	170.97	0.99
90%	172.79	1.28
98%	177.62	2.05

WBS with Estimation

名稱	工時
⊟ Meeting Scheduler	993
⊞ Survey	60
⊟ 專案規劃	30
發展WBS	10
發展專案計畫書	20
⊟ 需求分析	338
需求擷取	25
訂定目標	20
定義角色	15
⊞ 發展使用案例	166
發展Data Dictionary	21
發展系統架構	21
撰寫需求文件	70
M1：需求文件產出	
⊞ 系統設計	257
⊞ 系統實作	210
⊞ 系統測試	95
M6：驗收測試	

Scheduling

- Schedule development is the next step of project estimation
  - Based on estimated effort and the dependencies between tasks
- Scheduling Methods
  - Critical Path Method (CPM)
    - Single early/late start and finish date for all activities. Based on specified, sequential network and single duration estimate. Calculates float to determine flexibility.
  - Program Evaluation and Review Technique (PERT)
    - Sequential network and weighted average duration to calculate project duration, differs from CPM by taking into consideration of **optimistic (best case), pessimistic (worst case), and most-likely estimates.**

Task Dependency

- There are 3 primary types of dependencies:
  - Mandatory Dependencies
    - **Firm** relationships between two activities.
    - an activity cannot start until another activity finishes.
  - Discretionary Dependencies
    - **Soft** dependencies that reflect how the team anticipates the work will be completed, or how the team “would like” the work to be completed.
    - an activity will start up a couple weeks after another activity is completed.
  - External Dependencies
    - Dependencies that are **outside the control** of the project team, but nonetheless must be reflected in the project schedule.
    - approval from an external organization must be received prior to starting an activity.

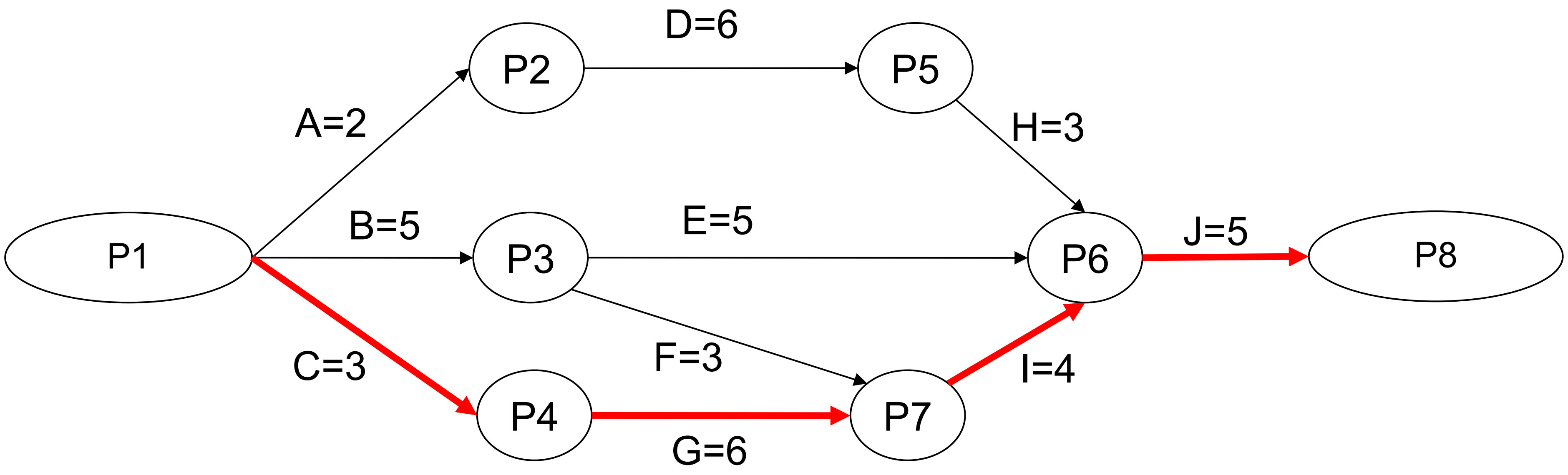
PERT Chart

- PERT chart is an activity-on-arrow graph.
  - The vertex (called event in PERT) marks the start or completion of one or more tasks and consumes no time and uses no resources. (**milestone**)
  - Each edge (called activity in PERT) is the actual performance of a task which consumes time (and requires resources).
- Activity estimation
  - There are three different types of time estimates per activity: Optimistic (O), Pessimistic (P), and Most Likely (M)
  - Expected time: (O + 4M + P)/6

Critical Path and Slack

- Critical path
  - The **longest possible continuous pathway** taken from the initial event to the terminal event.
  - It determines the total **calendar time** required for the project
- Slack (Float)
  - The amount of time **a task can be delayed without delaying the entire project. Tasks on critical path have no slack.**

PERT Chart



- The time required of the critical path *P1->P4->P7->P6->P8* is *18*.
- If the activity B is completed in *5 md* as planned, the slack (float) for activity E is *3 md* (*18-5-5-5=3*)

WBS with Schedule

名稱	工時	期間	起始日期	結束日期
Meeting Scheduler	993	304	2011/10/18	2012/12/15
Survey	60	30	2011/10/18	2011/11/29
Goal-driven	25	5	2011/11/22	2011/11/29
Meeting Scheduler Spec.	20	4	2011/10/18	2011/10/22
現有Meeting Scheduler系統	15	5	2011/10/24	2011/10/29
專案規劃	30	285	2011/11/14	2012/12/15
發展WBS	10	12	2011/11/14	2011/11/30
發展專案計畫書	20	272	2011/12/1	2012/12/15
需求分析	338	42	2011/10/24	2011/12/21
需求擷取	25	5	2011/10/24	2011/10/29
訂定目標	20	4	2011/11/29	2011/12/3
定義角色	15	3	2011/10/24	2011/10/27
發展使用案例	166	28	2011/11/3	2011/12/13
發展Data Dictionary	21	21	2011/11/14	2011/12/13
發展系統架構	21	21	2011/11/14	2011/12/13
撰寫需求文件	70	14	2011/11/29	2011/12/17
M1：需求文件產出		1	2011/12/20	2011/12/21

Task Schedule

Teamwork

- WBS can help a team perform better teamwork
  - Identify all tasks
  - Assign Responsibility
  - Facilitate communication and cooperation between project team and stakeholders
  - Get team buy-in, role identification
    - Project team members understand how their tasks fit into the overall project and their impacts upon the project

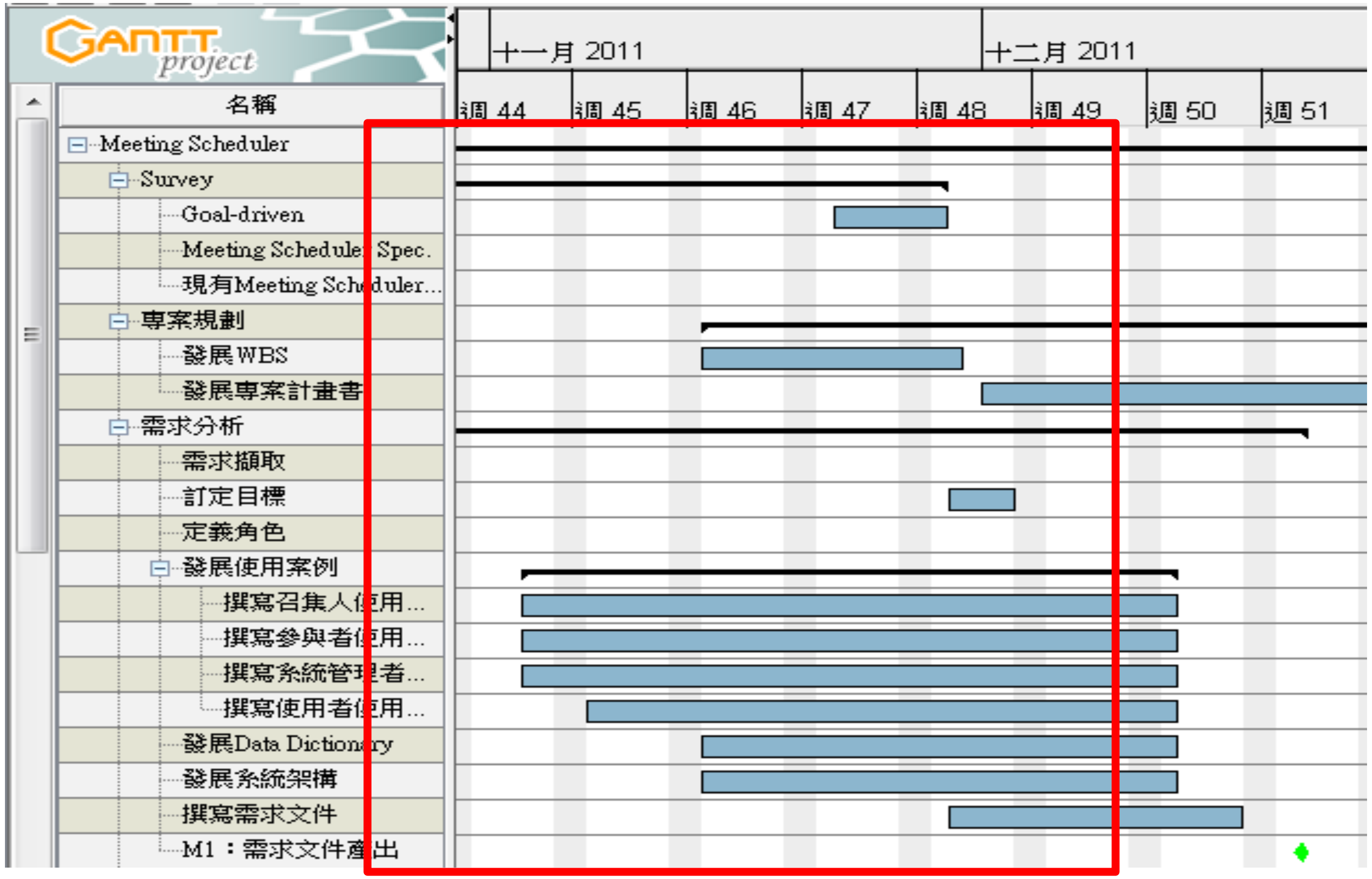
WBS with Assigned Responsibility

名稱	工時	期間	起始日期	結束日期	協調者
⊟ Meeting Scheduler	993	304	2011/10/18	2012/12/15	吳彥諄
⊟ Survey	60	30	2011/10/18	2011/11/29	吳彥諄
Goal-driven	25	5	2011/11/22	2011/11/29	吳彥諄
Meeting Scheduler Spec.	20	4	2011/10/18	2011/10/22	吳彥諄
現有Meeting Scheduler...	15	5	2011/10/24	2011/10/29	鄭聖翰, 洪東昇
⊟ 專案規劃	30	285	2011/11/14	2012/12/15	陳石佳
發展WBS	10	12	2011/11/14	2011/11/30	陳石佳
發展專案計畫書	20	272	2011/12/1	2012/12/15	陳石佳
⊟ 需求分析	338	42	2011/10/24	2011/12/21	吳彥諄
需求擷取	25	5	2011/10/24	2011/10/29	吳彥諄
訂定目標	20	4	2011/11/29	2011/12/3	吳彥諄
定義角色	15	3	2011/10/24	2011/10/27	吳彥諄
⊟ 發展使用案例	166	28	2011/11/3	2011/12/13	吳彥諄
撰寫召集人使用...	56	28	2011/11/3	2011/12/13	吳彥諄, 丘偉廷
撰寫參與者使用...	56	28	2011/11/3	2011/12/13	鄭聖翰, 洪東昇
撰寫系統管理者...	28	28	2011/11/3	2011/12/13	陳石佳
撰寫使用者使用...	26	26	2011/11/7	2011/12/13	丘偉廷
發展Data Dictionary	21	21	2011/11/14	2011/12/13	鄭聖翰
發展系統架構	21	21	2011/11/14	2011/12/13	吳彥諄
撰寫需求文件	70	14	2011/11/29	2011/12/17	吳彥諄
M1：需求文件產出		1	2011/12/20	2011/12/21	

Responsibility

Gantt Chart

- Gantt Charts is used to visualize task dependency, schedule and responsibility.



Typical WBS for Software Project

- Project Management
  - Project Execution Plan (PEP)
  - Milestone Review Record
  - Progress Review Record
- System Analysis
  - System Requirements Specification (SRS)
  - Interface Requirements Document
  - High Level System Architecture
- System Design
  - System Design Document (SDD)
  - Interface Design Document
  - Detail Level System Architecture
- System Implementation
  - Component 1, 2, 3 …
- System Integration & Test
  - XYZ Software v1.0 (version)
  - Test Plan and Procedure (TD)
  - Test Report
  - User Guide
  - Installation Guide
- Support (will be partially covered in Process Improvement Chapter)
  - PPQA Plan
  - MA Plan
  - CM Plan
  - PPQA Auditing Report
  - MA Report
  - CM Record

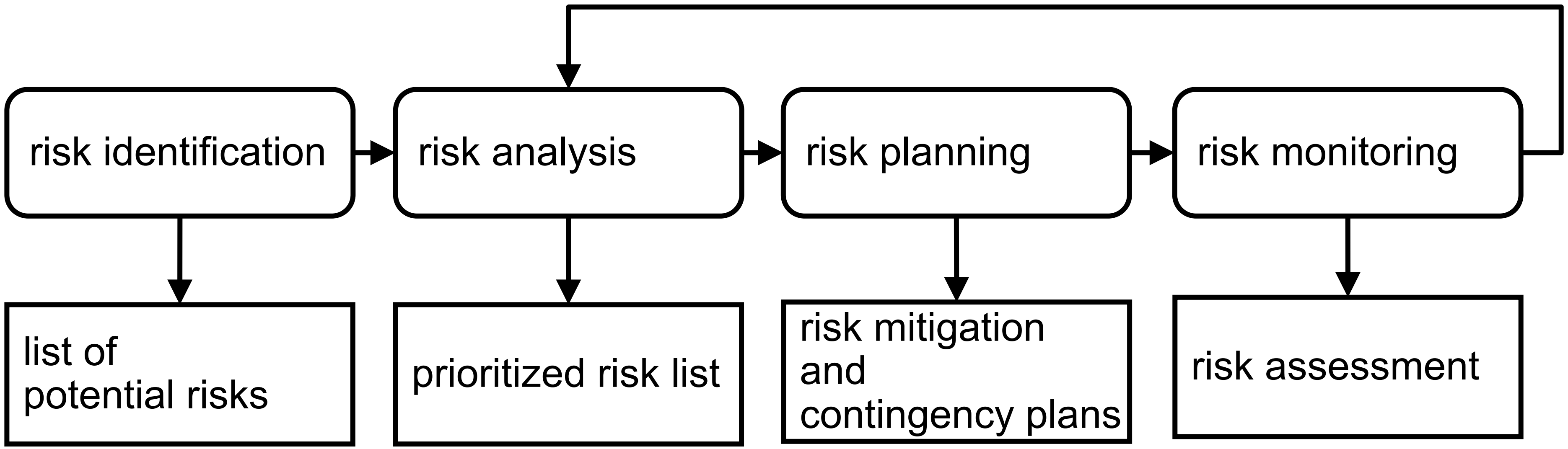
Risk Management

- Risk
  - The potential problems which may hinder the development of project.
  - The probability that some adverse event will occur.
    - Project risks affect schedule or resources;
    - Product risks affect the quality or performance of the software being developed;
    - Business risks affect the organization developing or procuring the software.
- Risk management: identify risks and draw up plans to minimize their effects on a project.

Risk Management Process

- Risk identification
  - Identify project, product and business risks
- Risk analysis
  - Assess the likelihood and consequences of these risks
- Risk planning
  - Draw up plans to avoid or minimize the effect of the risk
- Risk monitoring
  - Monitor the risks throughout the project

Risk Management Process



Risks Identification

Risk	Affects	Description
Staff turnover	Project	Experienced staff will leave the project before it is finished.
Management change	Project	There will be a change of organisational management with different priorities.
Hardware unavailability	Project	Hardware that is essential for the project will not be delivered on schedule.
Requirements change	Project and product	There will be a larger number of changes to the requirements than anticipated.
Specification delays	Project and product	Specifications of essential interfaces are not available on schedule
Size underestimate	Project and product	The size of the system has been underestimated.
CASE tool under-performance	Product	CASE tools which support the project do not perform as anticipated
Technology change	Business	The underlying technology on which the system is built is superseded by new technology.
Product competition	Business	A competitive product is marketed before the system is completed.

Risk Analysis

- Assess **probability and seriousness of each risk.**
  - Probability may be very low, low, moderate, high or very high.
  - Risk impacts (effects) might be catastrophic, serious, tolerable or insignificant.
  - Risk exposure is the product of risk probability and risk impact

$$R = \sum_{i=1}^{n} P_i \times I_i$$

R is the total risk exposure
P_i and I_i are the probability and the impact of an identified risk, respectively

Risk Examples

Enter into a new domain or technology

Happens to reusing Open Source

Happens all the time

That's why we Use Percent Confident Estimation

Risk	Probability	Impacts
It is impossible to recruit staff with the skills required for the project.	High	Catastrophic
Software components that should be reused contain defects which limit their functionality.	Moderate	Serious
Changes to requirements that require major design rework are proposed.	Moderate	Serious
The time required to develop the software is underestimated.	High	Serious
The organisation is restructured so that different management are responsible for the project.	High	Serious

Risk Planning

- Consider each risk and develop mitigation and/or contingency plans to manage the risks.
- Mitigation plans
  - Avoidance strategies
    - The probability that the risk will arise is reduced
  - Minimization strategies
    - The impact of the risk on the project or product will be reduced
- Contingency plans
  - If the risk arises, contingency plans are plans to deal with that risk

Risk Monitoring

- Assess each identified risk regularly to decide whether or not it is becoming less or more probable.
- Also assess whether the effects of the risk have changed.
- Each key risk should be discussed at progress review meetings.

A Risk Exposure Table

Impact levels \ Probability levels	Very Likely	Probable	Improbable
catastrophic	High	High	Medium
critical	High	Medium	Medium
marginal	High	Medium	Low
negligible	Medium	Medium	Low

High	Medium	Low

Threshold for triggering mitigation plans: Medium

Project Execution Plan (PEP)

- Introduction
- Project Lifecycle
- Work Breakdown Structure (WBS)
- Schedule
  - Milestone
- Resource
  - Personnel
    - Required Knowledge and Skill
    - Training Plan
  - Hardware/Software
- Risk Management
- Monitor and Control Mechanism

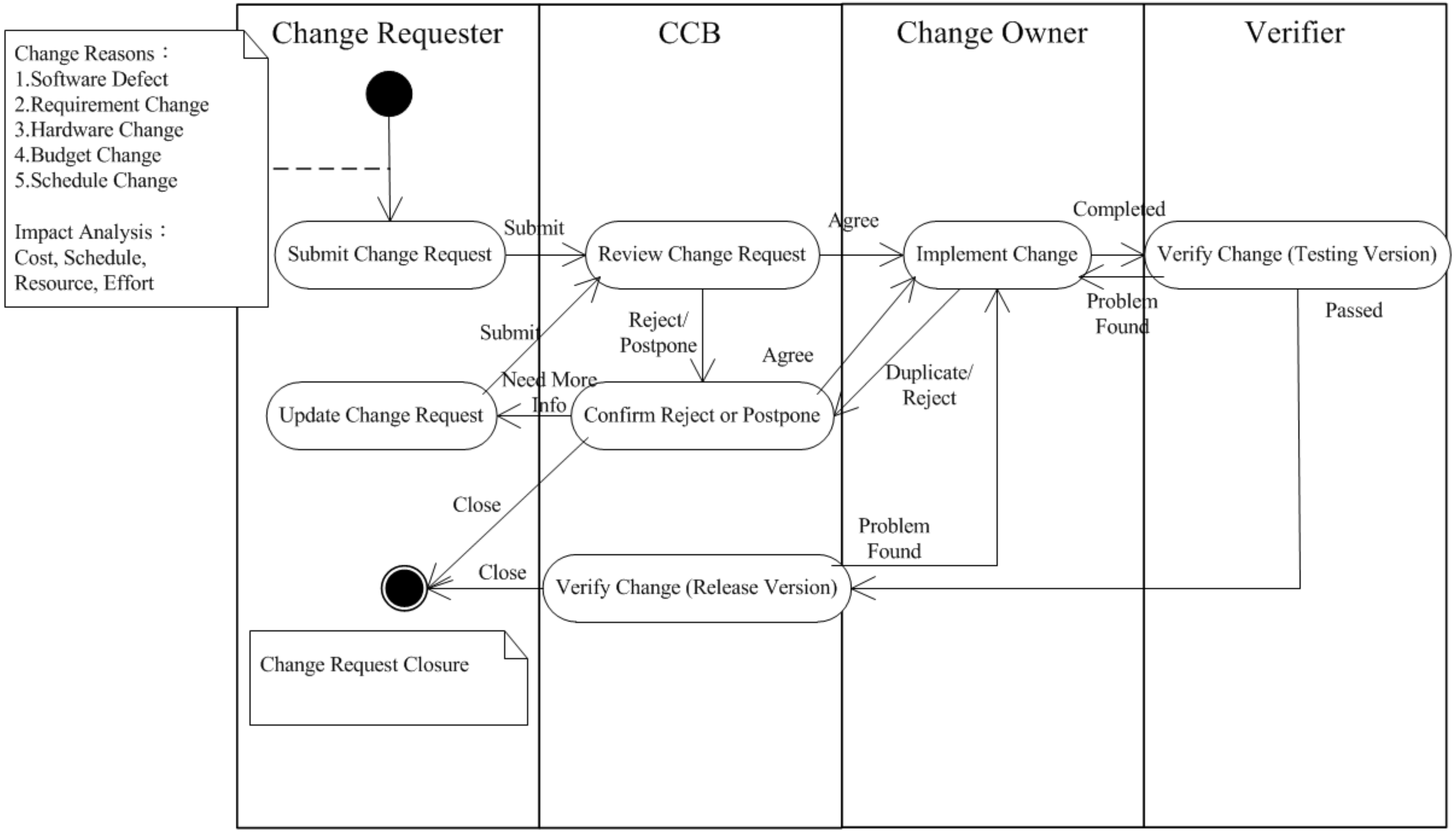
Executing Processes

- Team Development
  - Create a **captivating atmosphere** by encouraging discussion, co-operation, teamwork, interdependence and by building trust among team members.
- Complete Tasks/Work Packages
  - **Feasibility study (why is it so important?)**, knowledge and skill required for the tasks, don't ignore minor issues, **follow the processes (always)**.
- Scope Verification
  - Acceptance of project scope
  - Key aspect of scope verification is customer acceptance

Controlling Processes

- To regularly measure project performance and to adjust project plan.
- Change control mechanism (Change Control Board, CCB): Take preventive actions in anticipation of possible problems, or reactive actions to problems occurred
  - Change Control, Scope Change Control, Schedule Control, Cost Control, Quality Control, Risk Response Control
- Progress Review & Milestone Review.

Change Control Process



Scope Change Controlling

- Scope Change Control Inputs:
  - WBS
  - Issues reported
  - Change Requests – expansion/shrink of scope derived from :
    - External events (such as government regulations)
    - Scope definition errors of product or project
    - Value adding change (such as new technology)
- Scope Change Control Outputs:
  - Revised WBS
  - Corrective actions
  - Lessons Learned: cause and reasoning for variances

Monitoring and Control

- The project manager should monitor actual performance and progress of the project against the project plan (WBS is the core)
  - Progress review
  - Milestone review

Progress Review

- **Periodically review** the project's progress, performance, and issues.
  - Regularly communicate **status** on assigned activities and work products to relevant stakeholders.
  - Review the results of collecting and analyzing measures for controlling the project.
  - Identify and document significant **issues and deviations** from the plan.
  - Document **change requests**.
  - **Track action items to closure.**

Milestone Review

- Review the accomplishments and results of the project at selected project milestones.
  - Conduct reviews at meaningful points in the project's schedule, such as the completion of selected stages, with relevant stakeholders.
  - Review the commitments, plan, status, and risks of the project.
- Milestones can be event based or calendar based.
  - If the duration between two **event-based milestones is too long**, **calendar-based milestones can be inserted** to enhance the effect of project monitoring.

Meeting Minutes

- Meeting Location
- Meeting Start Time and End Time
- Attendance
- Agenda: Including Issues and risks
- Action Items
- Decisions Made
- Next Meeting

Issue

- Issues are major questions to be resolved
- Examples of issues to be gathered include the following:
  - Issues discovered when performing technical reviews, verification, and validation
  - Significant deviations in project planning parameters from estimates in the project plan
  - Commitments (either internal or external) that have not been satisfied
  - Significant changes in risk status
- Can be managed by Issue Tracking System
  - BugZilla, Trac, Mantis, etc.

Action Item

- Action Item: a task assigned to a person to be done by a certain time
  - What?, Who?, When?
  - Heuristics for Duration: be done within one week or two weeks
  - Generated from WBS or raised in the progress review meeting
  - Including corrective actions to address issues
- Action items should be tracked by the project manager

Action Item Tracking

Action Item 後續處理項目					
編號	處理動作	負責人員	處理期限	狀態	備註
1	Meeting Scheduler Spec. Survey	吳彥諄	2011/10/22	**Closed 10/22**	
2	現有 Meeting Scheduler 系統 Survey	鄭聖翰 陳石佳	2011/10/29	**Closed 10/29**	
3	發展 WBS	洪東昇	2011/11/30	**Closed 11/30**	
4	發展專案計畫書	洪東昇	2011/12/15	**Closed 12/15**	
5	寄出公司參訪感謝函及簡報	陳石佳	2011/12/29	**ongoing**	
6	企業參訪投影片製作	全體人員	2011/12/26	**Closed 12/26**	
7	編寫追蹤矩陣	鄭聖翰	2011/12/19	**Closed 12/19**	預計於 12/19 晚上整合完畢並審查
8	Goal-Driven Use Case 召集人、使用者 撰寫	丘偉廷 吳彥諄	2011/12/12	**Closed 12/12**	

From WBS

Raised in Meetings

Closing Processes

- Administrative Closure – generating necessary information to formally recognize phase or project completion
- Contract Close-out – completion and delivery of project deliverables and resolving open issues
  - Product Verification
  - Formal Acceptance
  - Lessons Learned
  - **Closure Report**

Project Closure Report

- Measurement and Analysis
  - Accuracy of Size Estimation
  - Accuracy of Effort Estimation
  - Defect Distribution
  - Defect Density
- Analysis of Process Execution
  - Evaluate all the processes involved in the project and raise problems encountered and further improvement suggestion.
  - CM, RD, REQM, TS, VER, VAL, PP, PMC, RSKM, PI and etc.
- Lessons Learned