

WORK BREAKDOWN STRUCTURE × LINEAR.APP

Menghubungkan konsep Manajemen Proyek dengan praktik pengelolaan proyek perangkat lunak

WBS

PROJECT

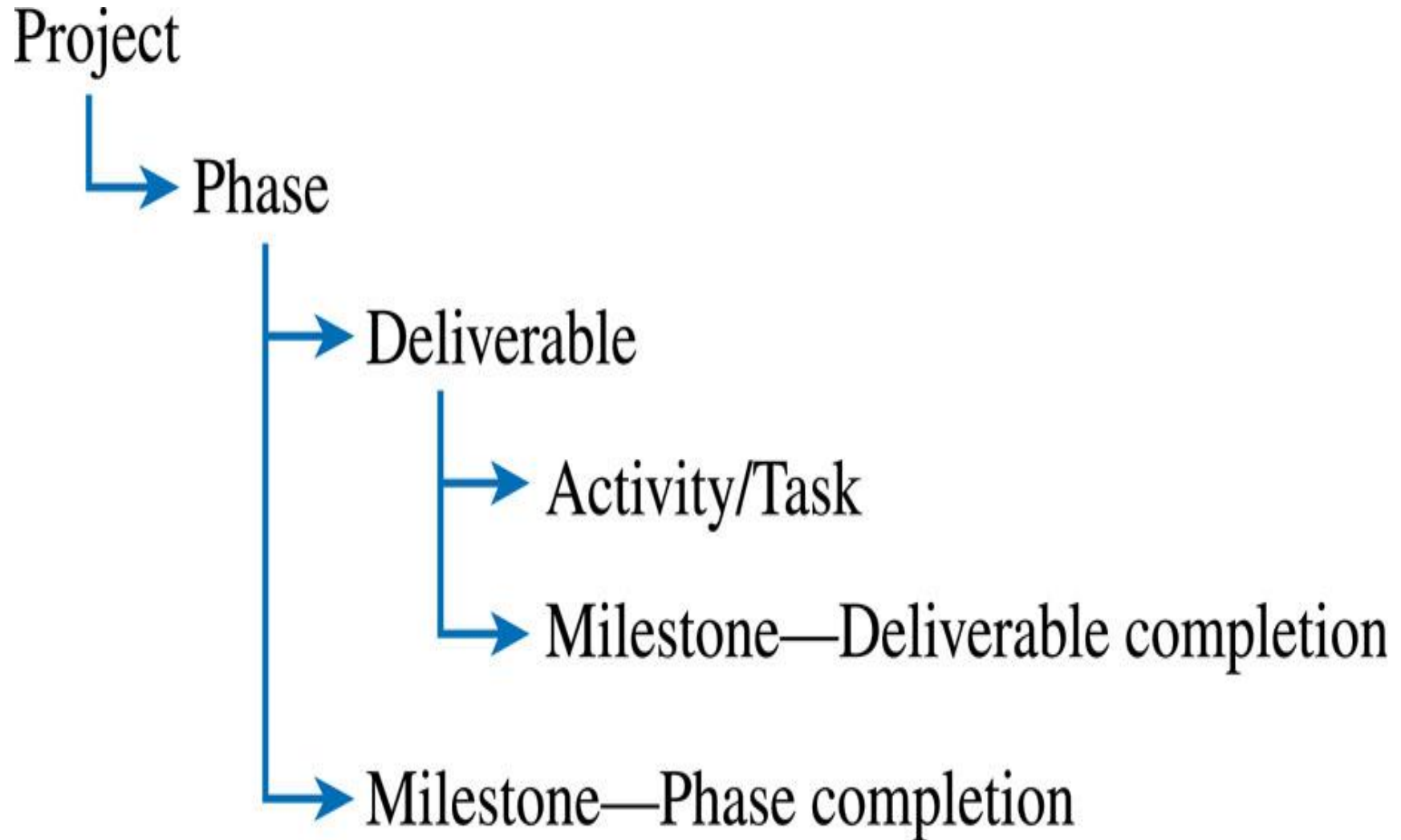
LINEAR.APP

Work Breakdown Structure (WBS)

- The WBS represents a logical decomposition of the work to be performed and focuses on how the product, service, or result is naturally subdivided. It is an outline of what work is to be performed.

LEVEL	ELEMENT DESCRIPTION
1	Project
2	Category
3	Subcategory
4	Sub-Subcategory
5	Work Package

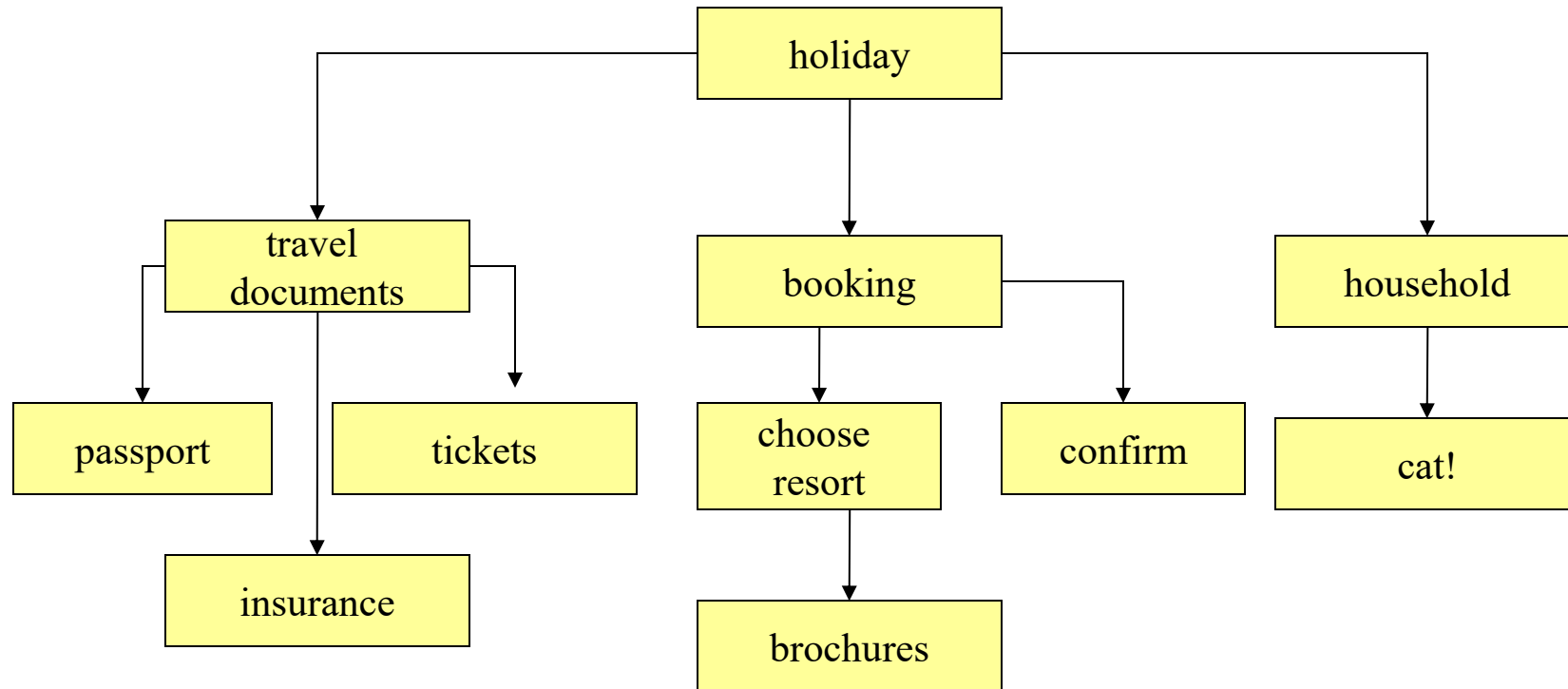
Work Package



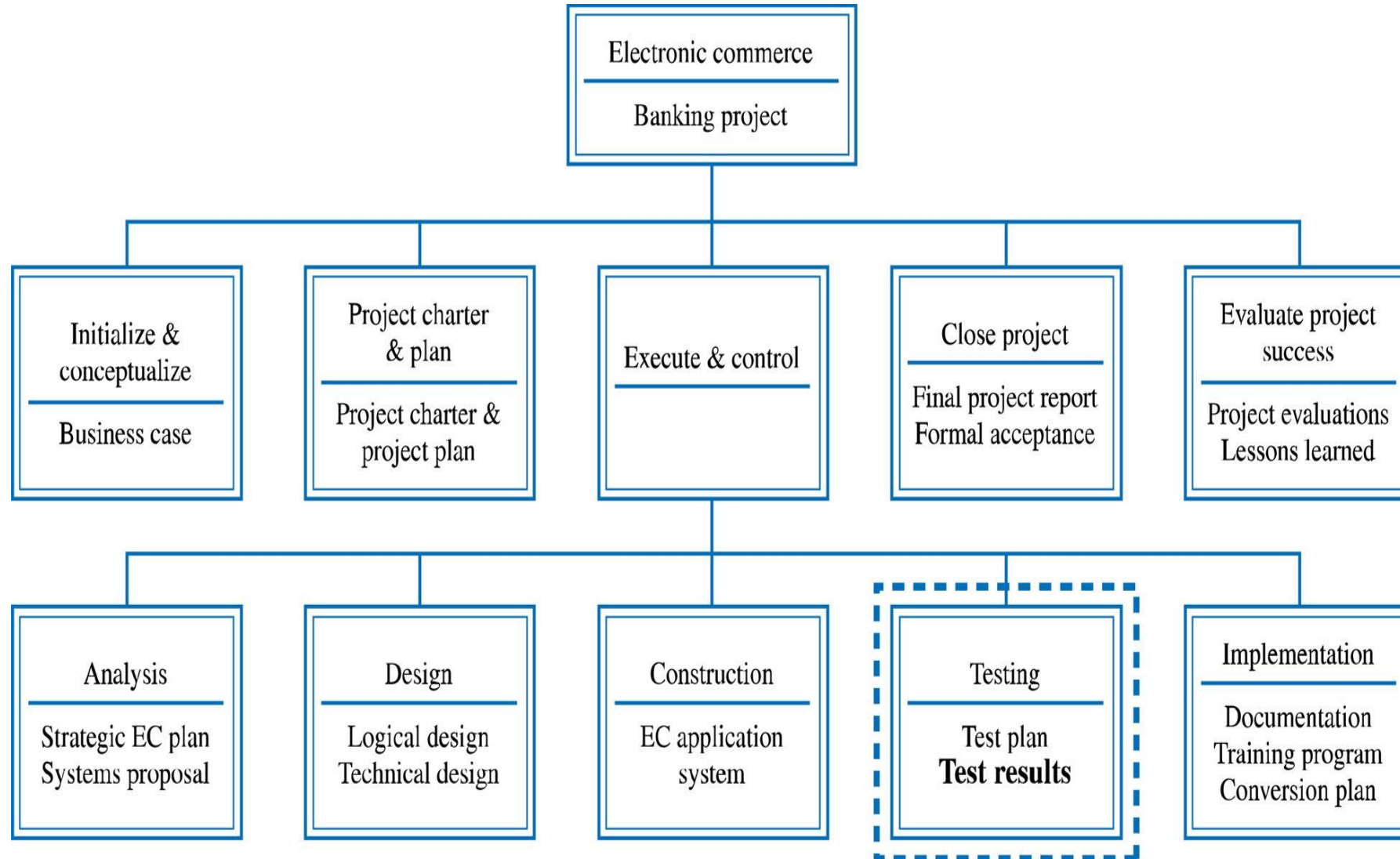
Deliverables and Milestones

- Deliverables
 - Tangible, verifiable work products
 - Reports, presentations, prototypes, etc.
- Milestones
 - Significant events or achievements
 - Acceptance of deliverables or phase completion
 - Cruxes (proof of concepts)
 - Quality control
 - Keeps team focused

Example of WBS: “Holiday”



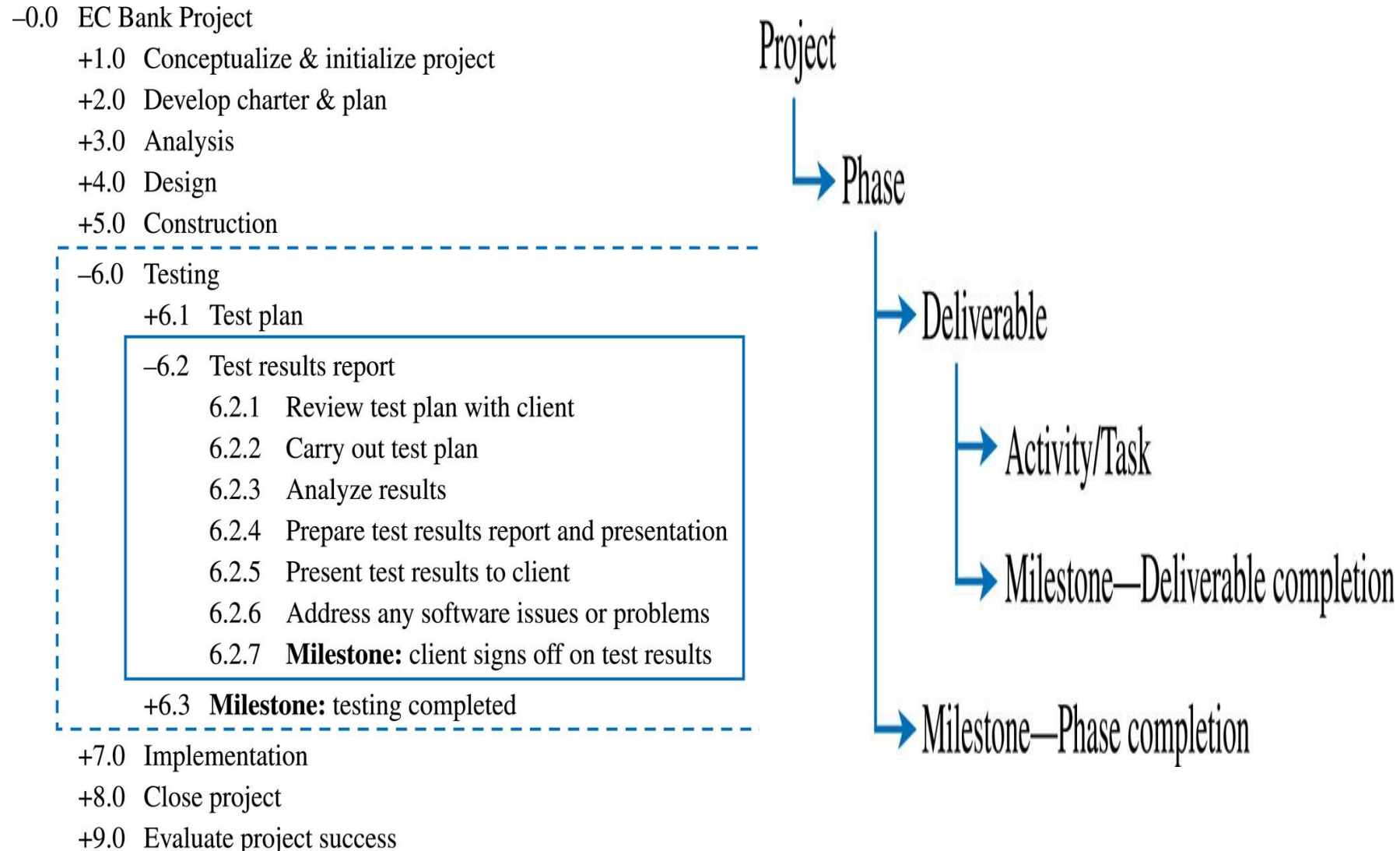
Developing the WBS



Example Work Breakdown Schedule

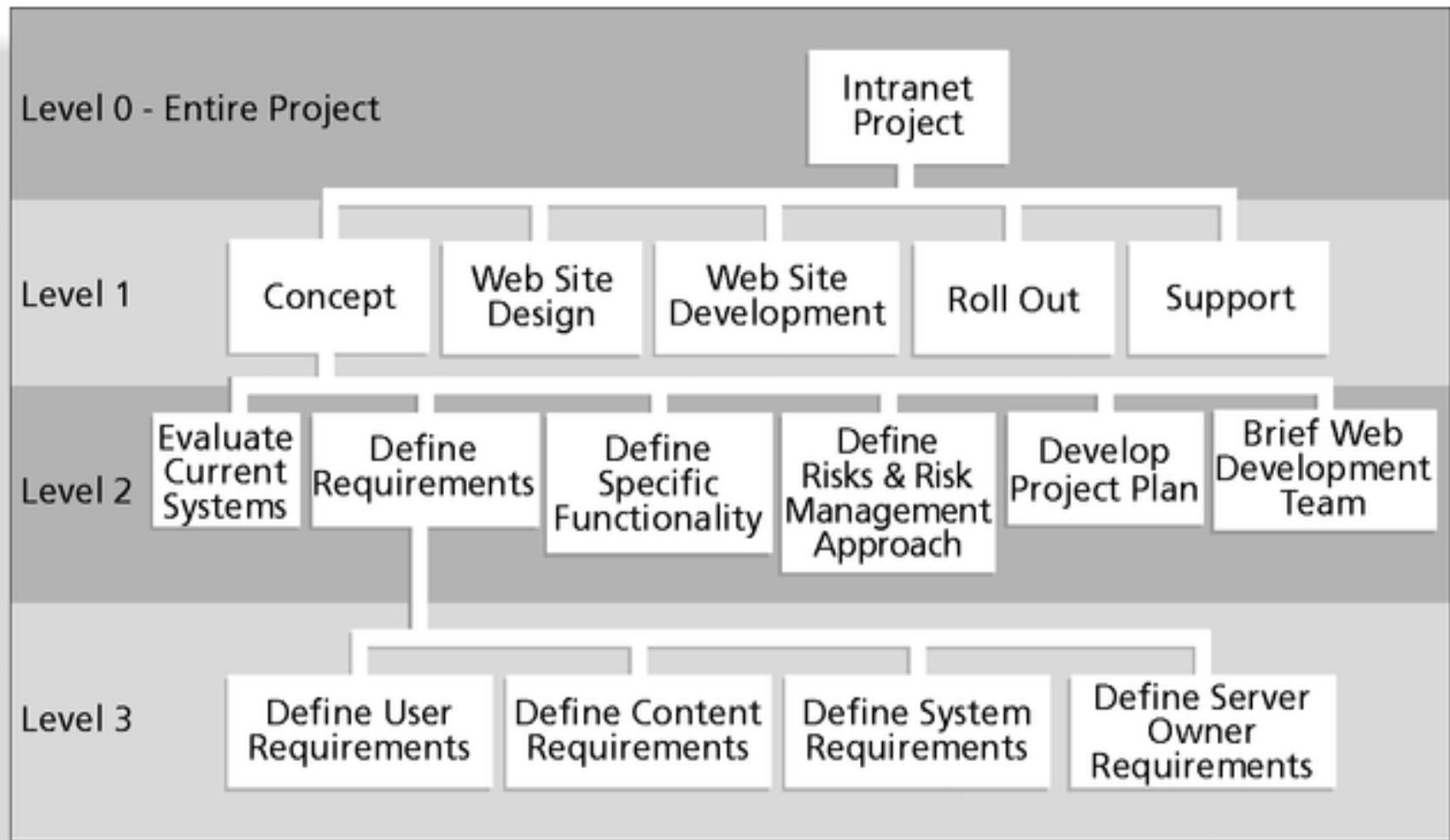
- 0.0 EC Bank Project
 - +1.0 Conceptualize & initialize project
 - +2.0 Develop charter & plan
 - +3.0 Analysis
 - +4.0 Design
 - +5.0 Construction
 - 6.0 Testing
 - +6.1 Test plan
 - 6.2 Test results report
 - 6.2.1 Review test plan with client
 - 6.2.2 Carry out test plan
 - 6.2.3 Analyze results
 - 6.2.4 Prepare test results report and presentation
 - 6.2.5 Present test results to client
 - 6.2.6 Address any software issues or problems
 - 6.2.7 **Milestone:** client signs off on test results
 - +6.3 **Milestone:** testing completed
 - +7.0 Implementation
 - +8.0 Close project
 - +9.0 Evaluate project success

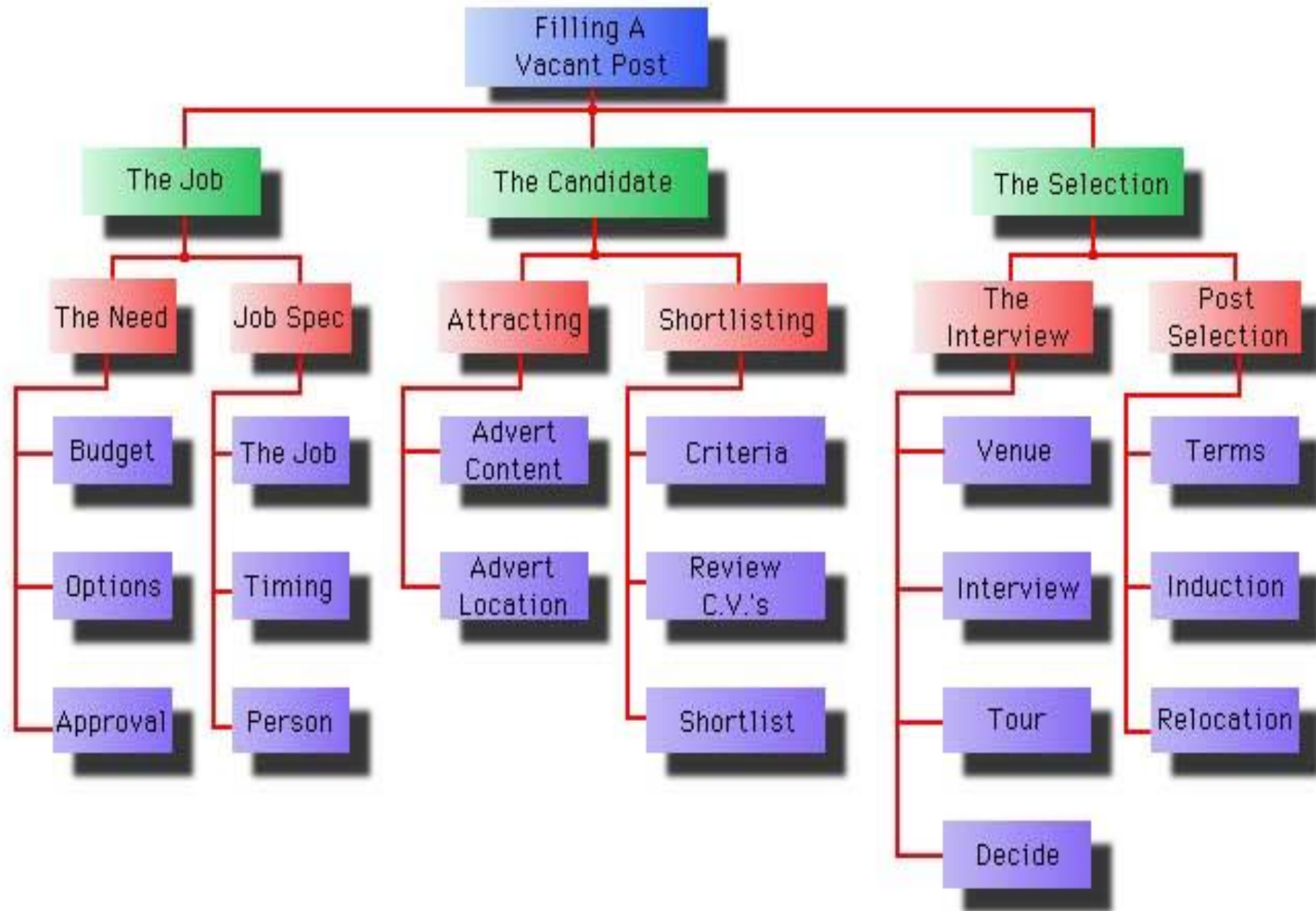
The WBS Should Follow the Work Package Concept

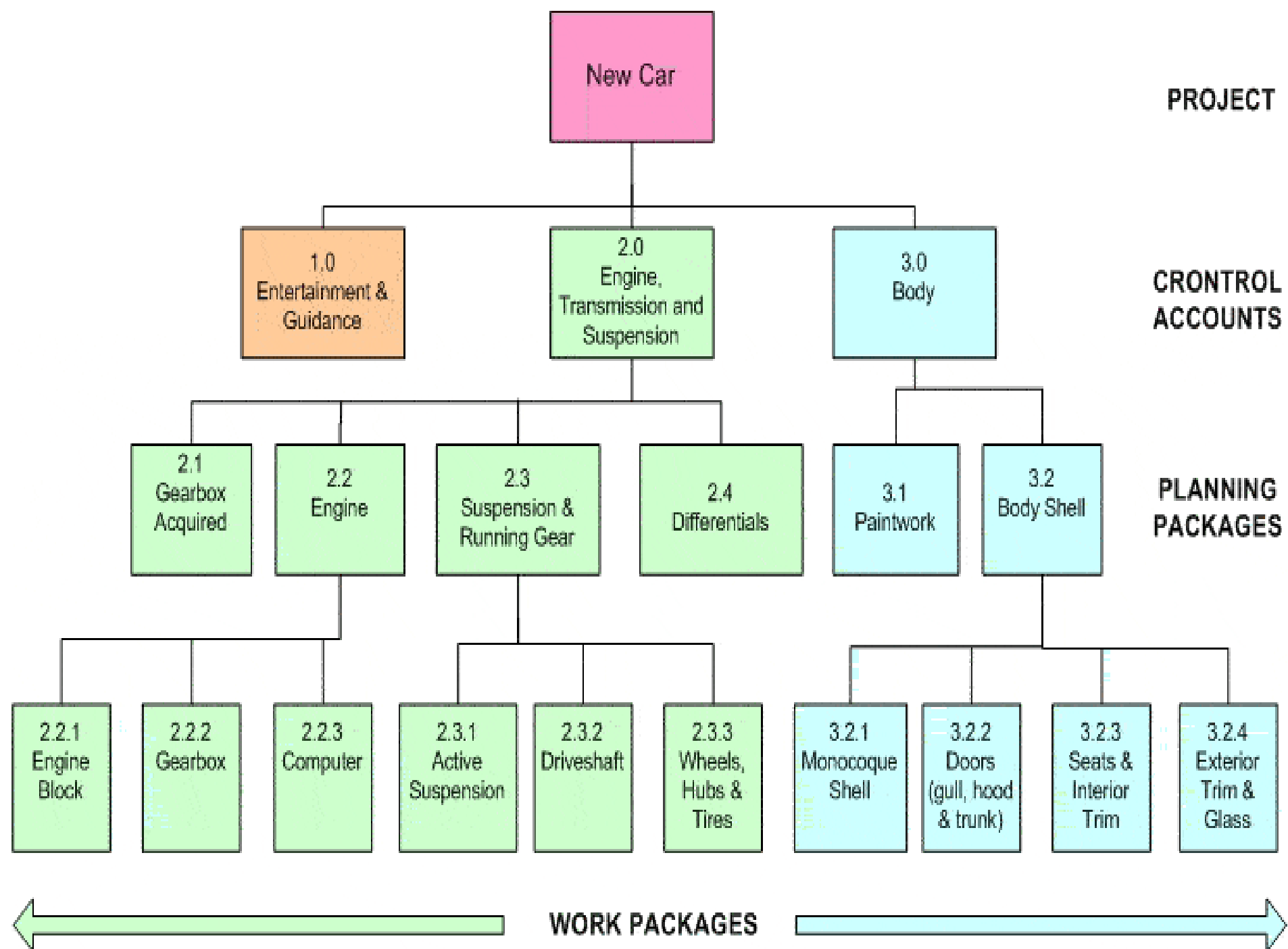


Developing the WBS

- The WBS Should Be Deliverable-Oriented
- The WBS Should Support the Project's MOV
 - Ensure WBS allows for the delivery of all the project's deliverables as defined in project scope
 - 100 percent rule
- The Level of Detail Should Support Planning and Control
- Developing the WBS Should Involve the People Who Will Be Doing the Work
- Learning Cycles and Lessons Learned Can Support the Development of a WBS







Responsibility Matrix

- Intersection of WBS and organization structure
- Rows = persons or functional positions
- Columns = work tasks or packages for which each personnel is responsible
- Useful for monitoring and control

Artifact	User Project Owner	System Analyst Project Manager	Software Architect	Use Case Specifier	Developer Roles	Test Roles	Administrator Roles	
Vision	A	G	G	W	G		G	G
Use Case Model	A	G	G	W				
Software Architecture Document	A				W		G	G
Use Case Specification	A	G		G	G	W		G
Use Case Realization					G	G	W	
Component					G		W	G

 = Write / responsible for
  = Contribute / review
  = Formally Accept

WBS		Responsibility					
		Project Office				Field Oper.	
Subproject	Task	Project Manager	Contract Admin.	Project Eng.	Industrial Eng.	Field Manager	
Determine need	A1	○		●	▲		
	A2	■	○	▲	●		
Solicit quotations	B1	○	■	▲		●	
Write approp. request.	C1	■	▲	○	●		
	C2		●	○	▲		
	C3	●	■	▲		■	
"	"						
"	"						
"	"						

Legend:

- ▲ Responsible
- Support
- Notification
- Approval

	Vice-president	General manager	Project manager	Manager engineering	Manager software	Manager manufacturing	Manager marketing	Subprogram manager manufacturing	Subprogram manager software	Subprogram manager hardware	Subprogram manager services
Establish project plan	6	2	1	3	3	3	3	4	4	4	4
Define WBS		5	1	3	3	3	3	3	3	3	3
Establish hardware specs		2	3	1	4	4	4				
Establish software specs		2	3	4	1		4				
Establish interface specs		2	3	1	4	4	4				
Establish manufacturing specs		2	3	4	4	1	4				
Define documentation		2	1	4	4	4	4				
Establish market plan	5	3	5	4	4	4	1				
Prepare labor estimate			3	1	1	1		4	4	4	4
Prepare equipment cost estimate		3	1	1	1			4	4	4	4
Prepare material costs			3	1	1	1		4	4	4	4
Make program assignments			3	1	1	1		4	4	4	4
Establish time schedules		5	3	1	1	1	3	4	4	4	4

1 Actual responsibility

2 General supervision

3 Must be consulted

4 May be consulted

5 Must be notified

6 Final approval

Estimation Techniques

- The Project Management Approach

- Guesstimating
- Delphi Technique
- Time Boxing
- Top-Down
- Bottom Up
- Analogous Estimates (Past experiences)
- Parametric Modeling (Statistical)

Project Estimation

- Guesstimating
 - Based on feeling and not facts
 - Not a good method for estimating but often used by inexperienced project managers
- Delphi Technique
 - Involves multiple, anonymous experts
 - Each expert makes an estimate
 - Estimates compared
 - If close, can be averaged
 - Another iteration until consensus is reached

Project Estimation

- Time Boxing
 - A “box” of time is allocated for a specific activity, task, or deliverable
 - Can focus a team if used effectively
 - Can demoralize a team if used too often or ineffectively because of the increased stress or pressure on the project team to get things done

Project Estimation

- Top-Down Estimating
 - Top and middle managers determine overall project schedule and/or cost.
 - Lower level managers are expected to breakdown schedule/budget estimates into specific activities (WBS).
 - Often couched in terms of what a project *should* cost and how long it *should* take as decreed by a member of top management who *thinks* those parameters are appropriate.
 - May be a response to the business environment.
 - May lead to a death march project.

Project Estimation

- Bottom-Up Estimating
 - Most common form of project estimation
 - Schedules & budgets are constructed from the WBS
 - Starts with people who will be doing the work
 - Schedules & budgets are the aggregate of detailed activities & costs

Project Estimation

- Analogous estimating
 - based on similarity between current projects and others
 - Use information from previous, similar projects as a basis for estimation

Project Estimation

- Parametric Modeling
 - Use project characteristics (parameters) in a mathematical model to estimate
 - Example: \$50/ LOC based on:
 - Programming language
 - Level of expertise
 - Size & complexity

WORK BREAKDOWN STRUCTURE × LINEAR.APP

Menghubungkan konsep Manajemen Proyek dengan praktik pengelolaan proyek perangkat lunak

WBS

PROJECT

LINEAR.APP

Dua Konsep yang Berbeda

WBS = struktur pekerjaan | Linear = sistem pengelolaan pekerjaan

WBS

**“APA SAJA PEKERJAAN YANG HARUS
DILAKUKAN?”**

LINEAR.APP

**“BAGAIMANA PEKERJAAN DIKELOLA &
DIPANTAU?”**

**WBS menentukan struktur pekerjaan → Linear
membantu menugaskan, menjadwalkan, mengurutkan,
dan memantau pekerjaan.**

Struktur WBS: Besar → Kecil

Contoh: Pengembangan Sistem Akademik



Mapping Utama: WBS ↔ Linear.app

Manajemen Proyek	Linear.app	Keterangan
Project	Project	Level proyek
Deliverable / Tahapan	Milestone	Target/tahapan penting
Work Package	Issue	Unit pekerjaan
Sub Work Package	Sub-Issue	Pekerjaan lebih detail
Periode/Sprint	Cycle	Waktu, bukan level WBS
Tim Pelaksana	Team	Struktur pelaksana
Organisasi/Wadah	Workspace	Ruang kerja utama
Tujuan/Program Strategis	Initiative	Kumpulan project

Workspace, Team, Initiative

Ketiganya penting, tetapi bukan level WBS

WORKSPACE
Ruang kerja utama
organisasi/tim

TEAM
Kelompok kerja / fungsi

INITIATIVE
Tujuan/program strategis

- Workspace = wadah administrasi dan pekerjaan.
- Team = siapa/kelompok mana yang mengerjakan.
- Initiative = mengapa beberapa project dikelompokkan untuk tujuan lebih besar.
- Ketiganya memberi konteks pengelolaan, bukan urutan pemecahan WBS.

PROJECT = Titik Temu Utama dengan WBS

**WBS
PROJECT**
Pengembangan Sistem Akademik

**LINEAR
PROJECT**
Sistem Akademik

Project di Linear menjadi wadah untuk outcome, target, milestone, dan issue yang membentuk pekerjaan proyek.

MILESTONE = Tahapan / Deliverable Penting

- Dapat digunakan untuk mengelompokkan pencapaian penting dalam project.
- Dalam WBS, paling dekat dengan deliverable atau tahapan besar.
- Contoh: Analisis Kebutuhan → Perancangan → Implementasi → Pengujian.
- Bukan seluruh detail WBS; pekerjaan detail direpresentasikan sebagai issue/sub-issue.

ANALISIS

DESAIN

IMPLEMENTASI

TESTING

ISSUE = Work Package / Pekerjaan yang Dapat Ditugaskan

WBS
3.1 Modul Login

LINEAR
Issue: Membuat Modul Login

OUTPUT
Pekerjaan konkret

- Issue dapat diberi assignee, priority, label, status, project, dan informasi lain.
- Contoh: database user; API login; halaman login; testing login.
- Issue sebaiknya spesifik sehingga tanggung jawab dan hasil akhirnya jelas.

SUB-ISSUE = Pemecahan WBS ke Tingkat Lebih Detail

ISSUE: Membuat Fitur Checkout

Desain Checkout

Database Orders

API Checkout

Halaman Checkout

Testing Checkout

Parent Issue → Sub-Issues = memecah pekerjaan besar menjadi bagian yang lebih terkelola.

CYCLE = WAKTU, BUKAN LEVEL WBS

Ini bagian yang paling penting untuk menghindari salah pemahaman

WBS
APA yang harus dikerjakan?

CYCLE
KAPAN dikerjakan?

Cycle/Sprint mengelompokkan sejumlah Issue berdasarkan periode kerja. Ia bukan anak dari Issue dan bukan level hierarki WBS.

Gambaran Lengkap: WBS + Linear.app



Contoh WBS: Sistem Akademik

- 0. Pengembangan Sistem Akademik
 - 1. Analisis Kebutuhan
 - 1.1 Identifikasi Stakeholder | 1.2 Kebutuhan Fungsional | 1.3 Kebutuhan Non-Fungsional
 - 2. Perancangan Sistem
 - 2.1 Database | 2.2 UI/UX | 2.3 Arsitektur
 - 3. Implementasi
 - 3.1 Modul Login | 3.2 Modul Mahasiswa | 3.3 Modul Dosen | 3.4 Modul KRS
 - 4. Pengujian
 - 4.1 Unit Testing | 4.2 Integration Testing | 4.3 UAT

Menerjemahkan WBS ke Linear.app

WBS 0: Sistem Akademik

WBS 1: Analisis

WBS 1.1: Stakeholder

WBS 2.2: UI/UX

WBS 3.1: Modul Login

WBS 3.1.1: Form Login

Project: Sistem Akademik

Milestone: Analisis Kebutuhan

Issue: Identifikasi Stakeholder

Issue: Perancangan UI/UX

Issue: Membuat Modul Login

Sub-Issue: Membuat Form Login

Contoh: WBS + Cycle

CYCLE 1 • 2 MINGGU

Analisis kebutuhan

Use case

Wireframe login

Database user

API autentikasi

Testing login

WBS menentukan pekerjaan; Cycle memilih pekerjaan mana yang dikerjakan dalam periode tertentu.

Alur dari Materi Kuliah ke Praktikum

1

Buat WBS

Pecah pekerjaan

2

Buat Project

Masukkan proyek

3

Milestone

Kelompokkan tahapan

4

Issue

Terjemahkan work package

5

Assign

Tentukan penanggung jawab

6

Cycle

Pilih periode

7

Status

Pantau progres

8

Review

Evaluasi hasil

Jangan Salah Memahami Hierarki

✗ KURANG TEPAT

Workspace → Team → Project → Milestone → Cycle →
Issue

Cycle dianggap sebagai level WBS.

✓ LEBIH TEPAT

Workspace / Team = konteks
Project / Milestone / Issue = struktur pekerjaan
Cycle = dimensi waktu

WBS adalah struktur hierarkis pekerjaan. Linear adalah alat untuk mengelola berbagai dimensi pekerjaan tersebut.

Latihan Kelas: CampusFood

- Kasus: Pengembangan Aplikasi Pemesanan Makanan Kampus.
- 1. Buat WBS minimal sampai level Work Package.
- 2. Tentukan 4 milestone.
- 3. Terjemahkan minimal 15 Work Package menjadi Issue.
- 4. Pilih 6–8 Issue untuk Cycle 1.
- 5. Buat minimal 3 Sub-Issue.
- 6. Tentukan assignee, priority, dan status.
- 7. Presentasikan hubungan WBS dengan struktur Linear.app.

Takeaway untuk Mahasiswa

Tiga pertanyaan kunci

**APA?
WBS**

**Pekerjaan apa yang harus
dilakukan?**

SIAPA?

TEAM / ASSIGNEE

Siapa yang mengerjakan?

KAPAN?

CYCLE

Kapan pekerjaan dikerjakan?

**PROJECT + MILESTONE + ISSUE + SUB-ISSUE menghubungkan struktur WBS dengan
eksekusi proyek.**

Ringkasan Istilah Linear.app

Workspace	Ruang kerja utama organisasi/tim	Konteks
Team	Kelompok kerja	Pelaksana
Initiative	Kumpulan project untuk tujuan strategis	Strategi
Project	Pekerjaan besar dengan outcome/target	Proyek
Milestone	Tahapan/pencapaian penting	Deliverable
Issue	Feature / bug / task	Work Package
Sub-Issue	Pekerjaan lebih detail	Sub Work Package
Cycle	Periode kerja berbasis waktu	Sprint / Waktu

Kesimpulan

Dari WBS menuju eksekusi proyek

- WBS memastikan seluruh pekerjaan proyek teridentifikasi dan terstruktur.
- Linear.app membantu mengubah struktur tersebut menjadi pekerjaan yang dapat ditugaskan, diprioritaskan, dijadwalkan, dan dipantau.
- Project, Milestone, Issue, dan Sub-Issue paling dekat dengan struktur pekerjaan.
- Workspace dan Team memberi konteks organisasi; Initiative memberi konteks strategis.
- Cycle adalah dimensi waktu/sprint untuk memilih pekerjaan yang dikerjakan pada periode tertentu.
- WBS + Linear membantu mahasiswa menghubungkan teori Manajemen Proyek dengan praktik Agile.

Referensi & Eksplorasi

Dokumentasi resmi dapat berubah mengikuti pembaruan aplikasi

- Linear — <https://linear.app/>
- Linear Documentation — <https://linear.app/docs>
- Linear Start Guide — <https://linear.app/docs/start-guide>
- Materi kuliah: Work Breakdown Structure (WBS), Work Package, Deliverable, Milestone, dan Scheduling.
- Praktikum lanjutan: Project → Milestone → Issue → Sub-Issue → Cycle → Review.

MENGGUNAKAN LINEAR.APP

Praktikum Manajemen Proyek Perangkat Lunak

- Linear — platform untuk merencanakan, melacak, dan mengelola pekerjaan pengembangan produk/perangkat lunak.
- Fokus praktikum: Workspace → Team → Project → Issue → Workflow → Cycle → Progress.
- URL: <https://linear.app/>

Tujuan Praktikum

- Membuat akun dan workspace Linear.
- Memahami konsep Team, Project, Issue, Milestone, dan Cycle.
- Membuat backlog dan memecah kebutuhan menjadi issue yang terukur.
- Mengatur workflow: Backlog → Todo → In Progress → In Review → Done.
- Mengelola sprint/cycle dan memantau progres tim.
- Memahami peluang integrasi Linear dengan GitHub/GitLab dan AI.

Konsep Dasar Linear

- Workspace: ruang kerja utama organisasi/tim.
- Team: kelompok kerja, misalnya Engineering, Design, atau Product.
- Project: pekerjaan besar dengan outcome dan target penyelesaian.
- Issue: unit pekerjaan terkecil seperti fitur, bug, atau task.
- Milestone: tahapan penting dalam sebuah project.
- Cycle: periode kerja berbasis waktu untuk mengatur pekerjaan berikutnya.
- Initiative: kumpulan project untuk tujuan organisasi yang lebih besar.

Struktur Manajemen Proyek

INITIATIVE — Tujuan strategis

PROJECT — Produk/fitur besar

MILESTONE — Tahapan pencapaian

ISSUE — Task / Bug / Feature

SUB-ISSUE — Pekerjaan lebih kecil

Manajemen Proyek (linear.app)

The screenshot displays the Linear app interface within a Chrome browser window. The browser's address bar shows the URL `linear.app/sularso/team/SUL/projects/all`. A notification banner at the top of the browser states, "Begin browsing instantly: Chrome can now launch when Windows starts. Allow".

The Linear app interface features a sidebar on the left with a user profile "Sularso Budilakso..." and a search icon. The sidebar menu includes "Agent", "Workspace", "Projects", "Views", "More", "Your teams", "Sularso Budilaksono", "Home", "Issues", "Projects" (highlighted), "Views", "Try", "Import issues", "Invite people", and "Connect GitHub".

The main content area is titled "Projects" and includes a "+ New project" button. Below the title is a tab labeled "All projects". The central part of the page features a 3D cube icon and the heading "Projects". A descriptive text states: "Projects are larger units of work with a clear outcome, such as a new feature you want to ship. They can be shared across multiple teams and are comprised of issues and optional documents." At the bottom of this section are two buttons: "Create new project" (with a sub-label "N then P") and "Documentation".

The Windows taskbar at the bottom shows the search bar with the text "Type here to search", several application icons (including Edge, File Explorer, and WhatsApp), and the system clock displaying "18:59" on "29/09/2026".

Praktikum 1 — Membuat Workspace & Team

- 1. Buka <https://linear.app/> lalu pilih Sign up / Log in.
- 2. Buat Workspace sesuai kelompok/proyek praktikum.
- 3. Buat atau gunakan Team, misalnya: Software Engineering.
- 4. Undang anggota kelompok sebagai member.
- 5. Tentukan role sederhana: Product Owner, Project Manager/Scrum Master, Developer, Tester/QA, UI/UX.
- Output: satu workspace dan satu team yang berisi seluruh anggota.

Praktikum 2 — Membuat Project

- Contoh project: 'Sistem Informasi Akademik Berbasis Web'.
- Tentukan Project Lead dan target penyelesaian.
- Tuliskan deskripsi/outcome project.
- Buat milestone, misalnya: Analisis → UI/UX → Development → Testing → Release.
- Gunakan project sebagai pusat informasi pekerjaan tim.
- Output: satu project dengan target, deskripsi, milestone, dan anggota.

Praktikum 3 — Membuat Issue

- Issue adalah unit pekerjaan utama di Linear.
- Contoh issue:
- 'Membuat desain halaman Login'
- 'Membuat API autentikasi'
- 'Membuat database tabel User'
- 'Testing validasi login'
- Setiap issue minimal memiliki title dan status; tambahkan assignee, priority, label, project, dan due date bila diperlukan.
- Gunakan issue yang spesifik dan dapat diselesaikan dalam waktu relatif singkat.

Praktikum 4 — Workflow & Kanban

- Gunakan workflow untuk menunjukkan perjalanan sebuah issue.
- Contoh: Backlog → Todo → In Progress → In Review → Done.
- Backlog: pekerjaan belum diprioritaskan untuk segera dikerjakan.
- Todo: pekerjaan siap dimulai.
- In Progress: sedang dikerjakan.
- In Review: menunggu review/testing.
- Done: pekerjaan selesai.

Praktikum 5 — Cycle / Sprint

- Buat Cycle 1 untuk periode kerja 1–2 minggu.
- Masukkan issue yang realistis untuk diselesaikan dalam cycle.
- Tetapkan prioritas dan assignee.
- Setiap anggota mengubah status issue sesuai progres.
- Pada akhir cycle, evaluasi issue selesai, belum selesai, dan hambatan.
- Gunakan hasil cycle sebagai dasar perencanaan cycle berikutnya.

Praktikum 6 — Issue Relations & Sub-Issues

- Gunakan parent/sub-issue untuk memecah pekerjaan besar.
- Gunakan relasi 'blocked by' ketika pekerjaan bergantung pada issue lain.
- Gunakan 'blocking' untuk menunjukkan issue yang menghambat pekerjaan berikutnya.
- Gunakan 'duplicate' ketika dua issue ternyata memiliki pekerjaan yang sama.
- Tujuan: dependency dan pembagian pekerjaan menjadi lebih jelas.

Praktikum 7 — Integrasi dengan GitHub / GitLab

- Hubungkan repository pengembangan dengan Linear.
- Gunakan ID issue pada branch atau pull request agar pekerjaan dapat ditelusuri.
- Status issue dapat dikaitkan dengan aktivitas pull request.
- Contoh alur: Linear Issue → Git Branch → Commit → Pull Request → Review → Merge → Done.
- Manfaat: hubungan antara pekerjaan manajemen proyek dan aktivitas coding lebih mudah dilacak.

Linear + AI

- Linear saat ini menyediakan fitur AI/Agents untuk membantu pekerjaan berbasis issue dan project.
- Contoh penggunaan: membuat issue dari percakapan, merangkum update project, membantu memeriksa status project, dan mengotomasi pekerjaan tertentu.
- Linear juga menyediakan MCP untuk menghubungkan tool AI yang kompatibel dengan issue, project, dan komentar.
- Dalam praktikum lanjutan: gunakan AI untuk membantu membuat backlog → issue → task → dokumentasi.

Mini Project Praktikum

- Kasus: Tim mahasiswa mengembangkan 'Aplikasi Pemesanan Makanan Kampus'.
- Buat 1 Workspace dan 1 Team.
- Buat 1 Project.
- Buat minimal 3 milestone.
- Buat minimal 15 issue yang mencakup Feature, Bug, UI/UX, Database, API, dan Testing.
- Buat minimal 2 Cycle.
- Terapkan workflow dan assignment.
- Tunjukkan dashboard/progress project pada saat presentasi.

Tugas Mahasiswa

- 1. Screenshot Workspace dan Team.
- 2. Screenshot Project beserta milestone.
- 3. Screenshot daftar issue dan workflow.
- 4. Screenshot Cycle/Sprint.
- 5. Jelaskan pembagian tugas setiap anggota.
- 6. Buat laporan singkat: bagaimana Linear membantu pengelolaan proyek perangkat lunak.
- 7. Presentasikan alur dari requirement sampai issue selesai.

Ringkasan

- Linear membantu tim mengorganisasi pekerjaan dari perencanaan hingga delivery.
- Project digunakan untuk pekerjaan besar; Issue untuk pekerjaan konkret.
- Workflow menunjukkan status pekerjaan.
- Cycle membantu tim bekerja dalam periode yang terstruktur.
- Relations dan sub-issues membantu mengelola dependency dan kompleksitas.
- Integrasi Git dan AI dapat memperluas Linear menjadi pusat koordinasi pengembangan perangkat lunak.

Referensi & Akses

Linear: <https://linear.app/>

Linear Start Guide: <https://linear.app/docs/start-guide>

Linear Docs: <https://linear.app/docs>

Linear Learn: <https://linear.app/learn>

Linear AI: <https://linear.app/ai>

Gunakan dokumentasi resmi sebagai acuan jika tampilan/menu Linear berubah.