

# 3 MITOS PENYEBAB KEGAGALAN PENERAPAN SCRUM

*Wahyu Utomo*

wahyu@brainmatics.com  
<http://brainmatics.com>  
08118228887





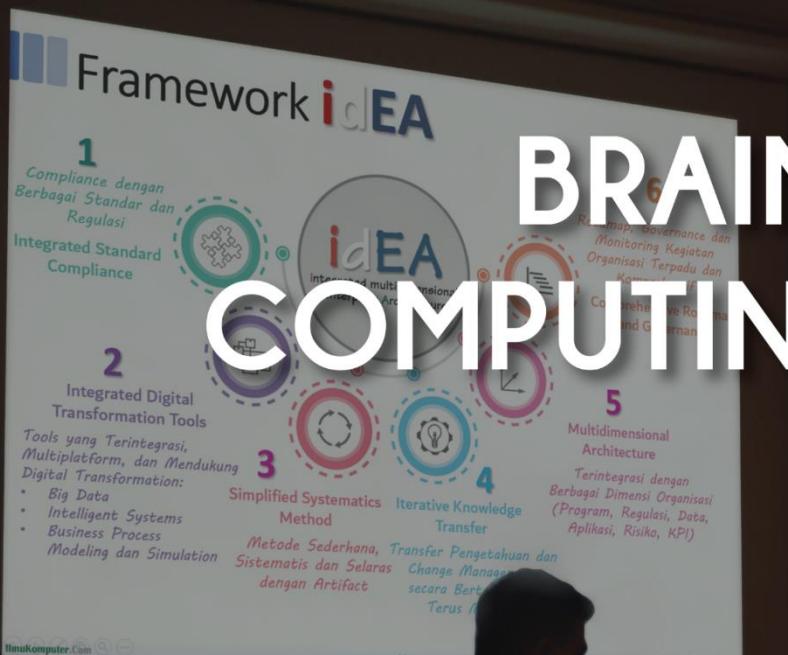
# Wahyu Utomo

- Direktur Marketing PT Brainmatics Indonesia Cendekia
- Project Manager dan Scrum Master PT IlmuKomputerCom  
Braindevs Sistema
- Enterprise Architecture Consultant: KPK, RistekDikti, LNSW,  
LIPI, Kemenkeu (Itjend, DJBC, DJPK), Kemsos, Telkom, PLN  
PJB, Pertamina (PEP, PPI) , ESDM, FIF, Kominfo (PPI,  
Balitbang), Diskominfo Kab. Kolaka, Diskominfo Tegal,etc
- Project Manager dan Scrum Master: Sistem Perisalah (BPPT),  
Portal Jasaraharja, Portal MINAMAS, KIAT GURU (TNP2K),  
FIFGROUP, etc.
- Industrial IT Certifications: Professional Scrum Master (PSM),  
TOGAF Foundation



# BrainmaticsID: Educating the Enterprise

# BRAINMATICS COMPUTING CURRICULA



**Romi Satria Wahono**

- SMA Taruna Nusantara Magelang (1993)
- B.Eng, M.Eng and Ph.D in Software Engineering Saitama University Japan (1994-2004)
- Universiti Teknkal Malaysia Melaka (2014)
- Core Competency in Enterprise Architecture, Software Engineering and Machine Learning
- LIPI Researcher (2004-2007)
- Founder, CoFounder and CEO:
  - PT Brainmatics Cipta Informatika (2005)
  - PT Imani Prima (2007)
  - PT IlmuKomputer.Com Braindevs Sistema (2014)
  - PT Brainmatics Indonesia Cendekia (2020)
- Professional Member of IEEE, ACM and PMI
- IT and Research Award Winners from WSIS (United Nations), Kemdikbud, Journal Reviewer: Information and Software Technology, and Software, Software: Practice and Experience, etc
- Publications: TOGAF, ITIL, CCAI, CCNA, etc
- Software & Digital Transformation Expert: KPK, INSW, BPPT, LIPI, Pertamina EP, PLN, PJB, PJBI, IP, FIF, Kemiu, ESDM, etc.

**BRAINMATICS**  
*The School of Computing*

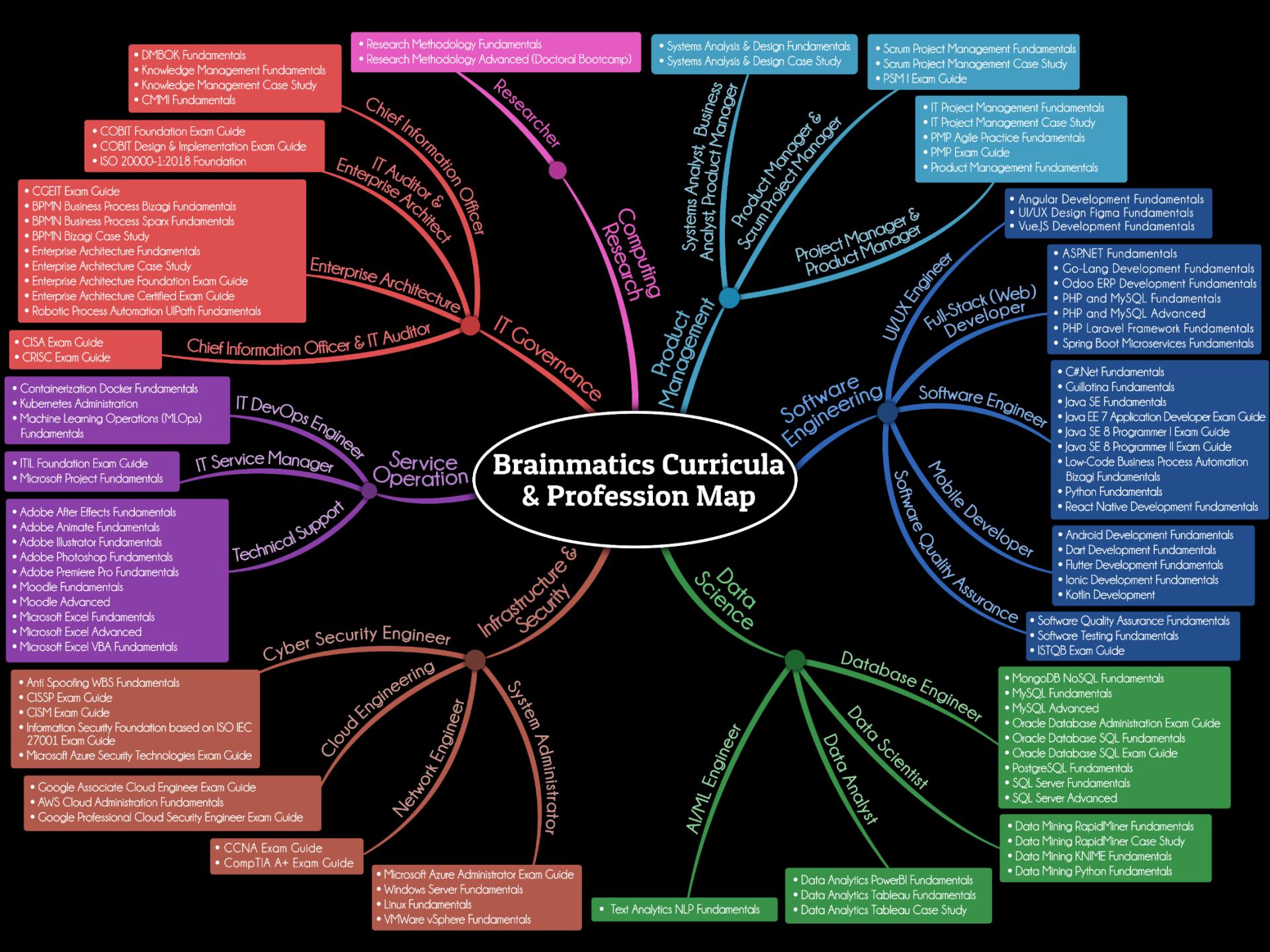
PT Brainmatics Indonesia Cendekia

Menara Bidakara 1 Lantai 2, Jl. Gatot Subroto Kav 71-73 Jakarta (12870)

info@brainmatics.id +622183793383 <https://brainmatics.id>



Gabung BrainmaticsID Community <https://t.me/BrainmaticsID>



## MITOS 1

Penerapan Scrum yang Masih  
**Bercampur dengan SDLC** Membuat  
Developers masih Mengikuti  
**Metodologi Waterfall**

# 3 MITOS

PENYEBAB KEGAGALAN

**PENERAPAN SCRUM**

## MITOS 3

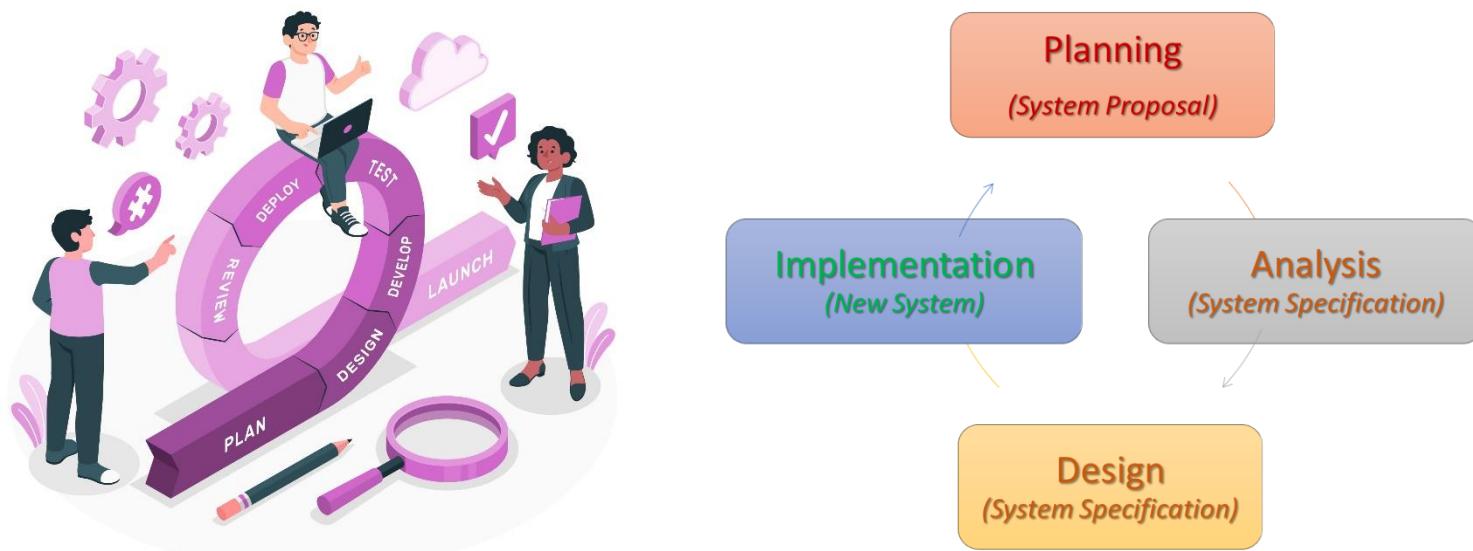
Masih Memikirkan  
**Dokumentasi Pengembangan**,  
dan Kurang Fokus ke  
Pengembangan Produk

## MITOS 2

Tidak Mengikuti Panduan yang  
sudah Dibuat Scrum Guide dan  
Malah Membuat Improvement  
yang Akhirnya Gagal

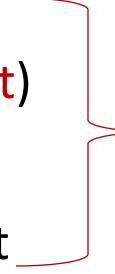
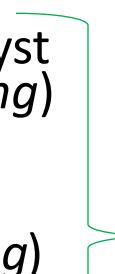
# MITOS 1

Penerapan Scrum yang Masih **Bercampur dengan SDLC** Membuat Developers masih Mengikuti Metodologi Waterfall

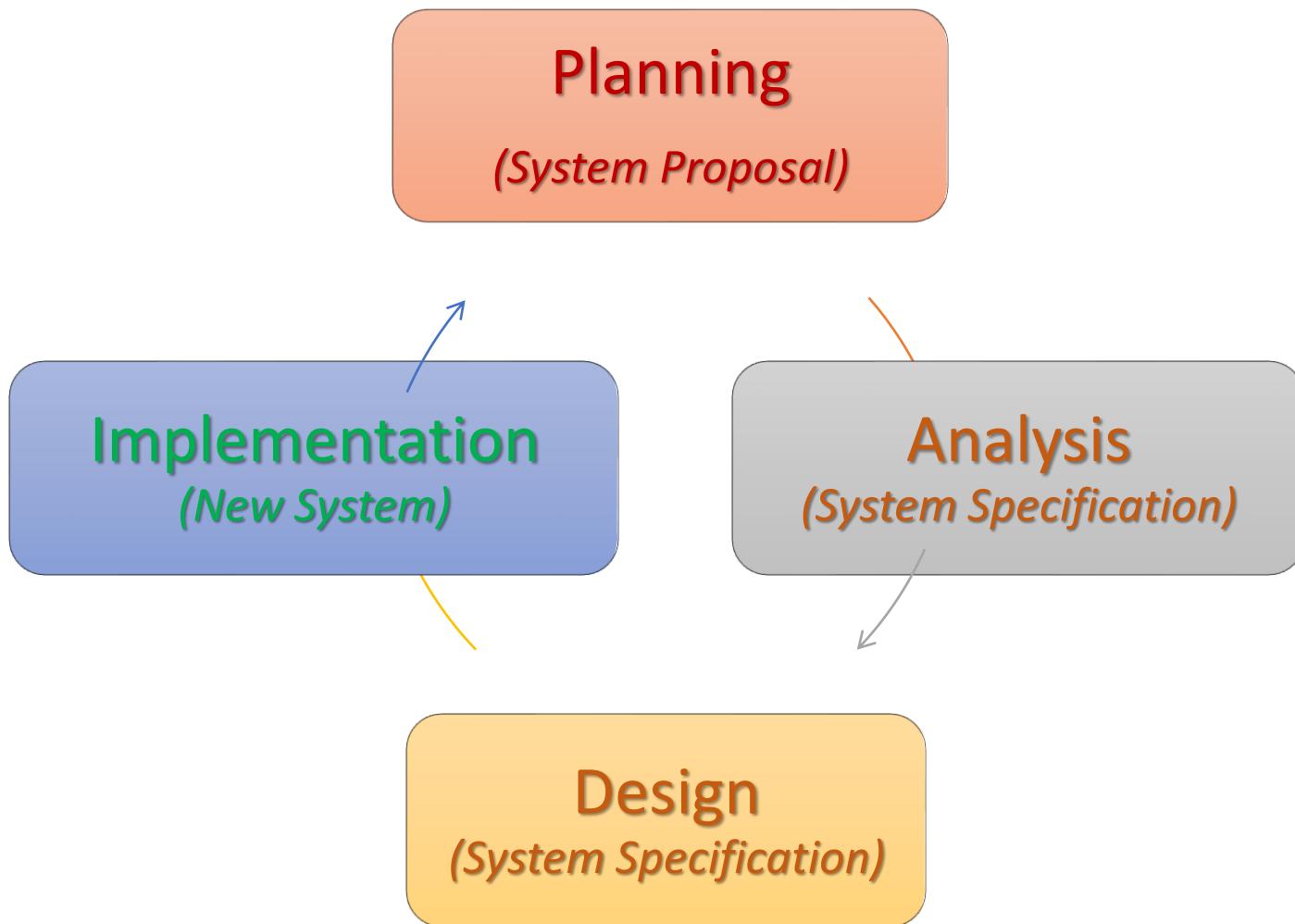


# Siklus Pengembangan Software:

Alur, Peran, dan Tahapan (*Deliverable*) (Tilley, 2012) (Dennis, 2016) (Valacich, 2017)

1. **User/Product Owner** membawa permintaan kebutuhan (perubahan) software (**System Request**) ke **System Analyst**
2. **System Analyst** membuat analisis kelayakan (**Feasibility Analysis**) dari System Request tersebut
3. Setelah dinyatakan layak, **System Analyst** melakukan analysis dan design, dan hasilnya adalah **System Specification**
  - **Business Analyst** membantu **System Analyst** memahami proses bisnis dari software yang akan dibangun
4. **System Specification** diserahkan oleh **System Analyst** ke **Programmer** untuk dilakukan **Konstruksi (Coding)**
5. Hasil Konstruksi berupa **Kode Program** diserahkan ke **Software Tester** untuk dilakukan **Pengujian (Unit, Integration, System, User Acceptance Testing)**
6. Instalasi (*delivery*) software dan manajemen perubahan
  - **Software** = Kode Program + Dokumentasi (Pengembangan dan Penggunaan)
7. Siklus kembali ke 1 apabila ada permintaan perubahan (**Permintaan Perubahan Software**)

# Siklus Pengembangan Software (SDLC)



(Tilley, 2012)

(Dennis, 2016)

(Valacich, 2017)

# Major Software Development Methodologies

## 1. Structured Design

(**Prescriptive**) (1967- )

- Waterfall method
- Parallel development



More  
Prescriptive/  
Documentation

## 2. Rapid Application Development

(**Iterative**) (1985-)

- Phased Development
- Prototyping



More  
Adaptive/  
Communication

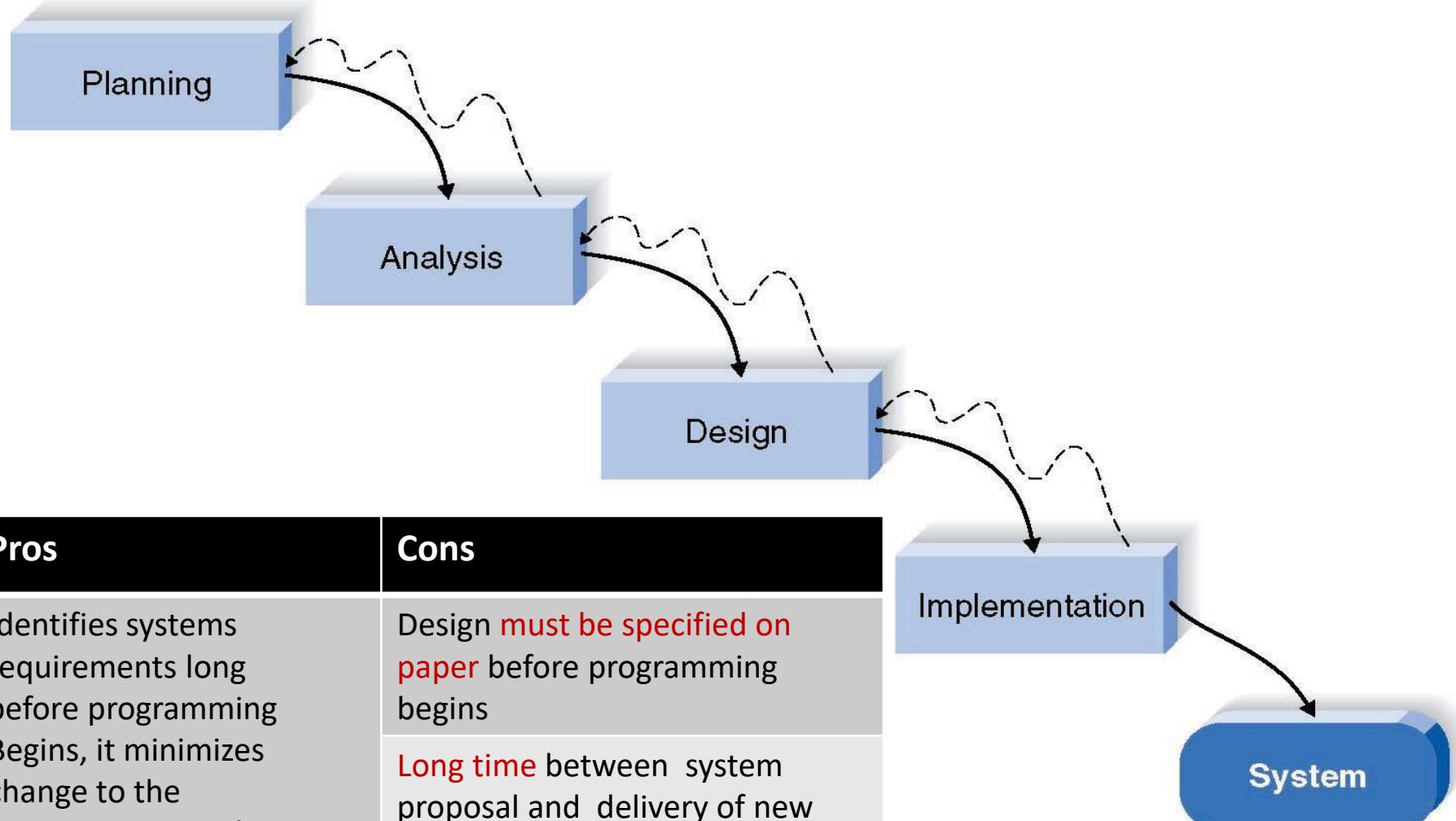
## 3. Agile Development

(**Adaptive**) (1995-)

- Extreme Programming (XP)
- Scrum

*Compiled from (Dennis, Wixom and Tegarden, 2016)*

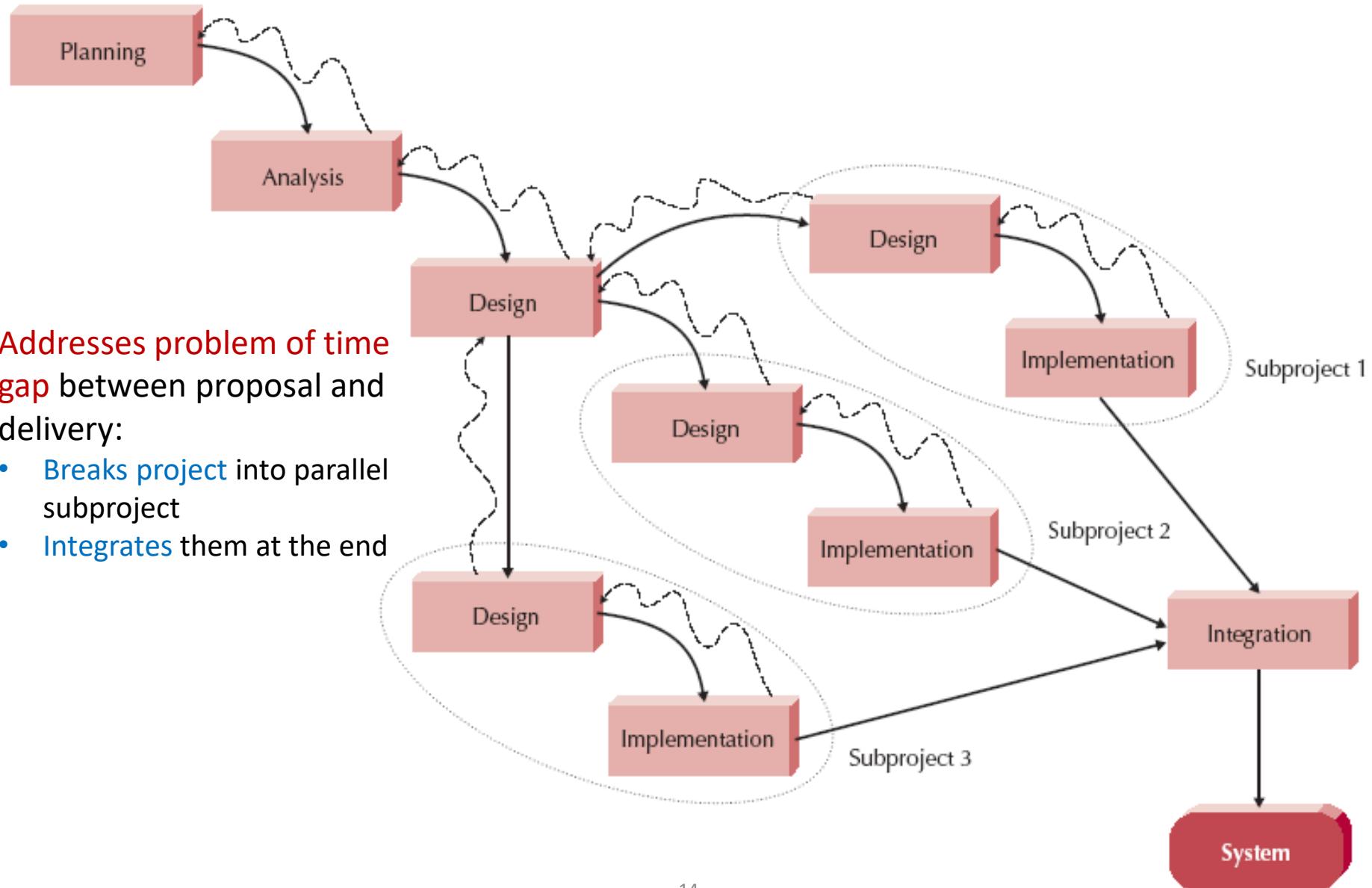
# Waterfall Method



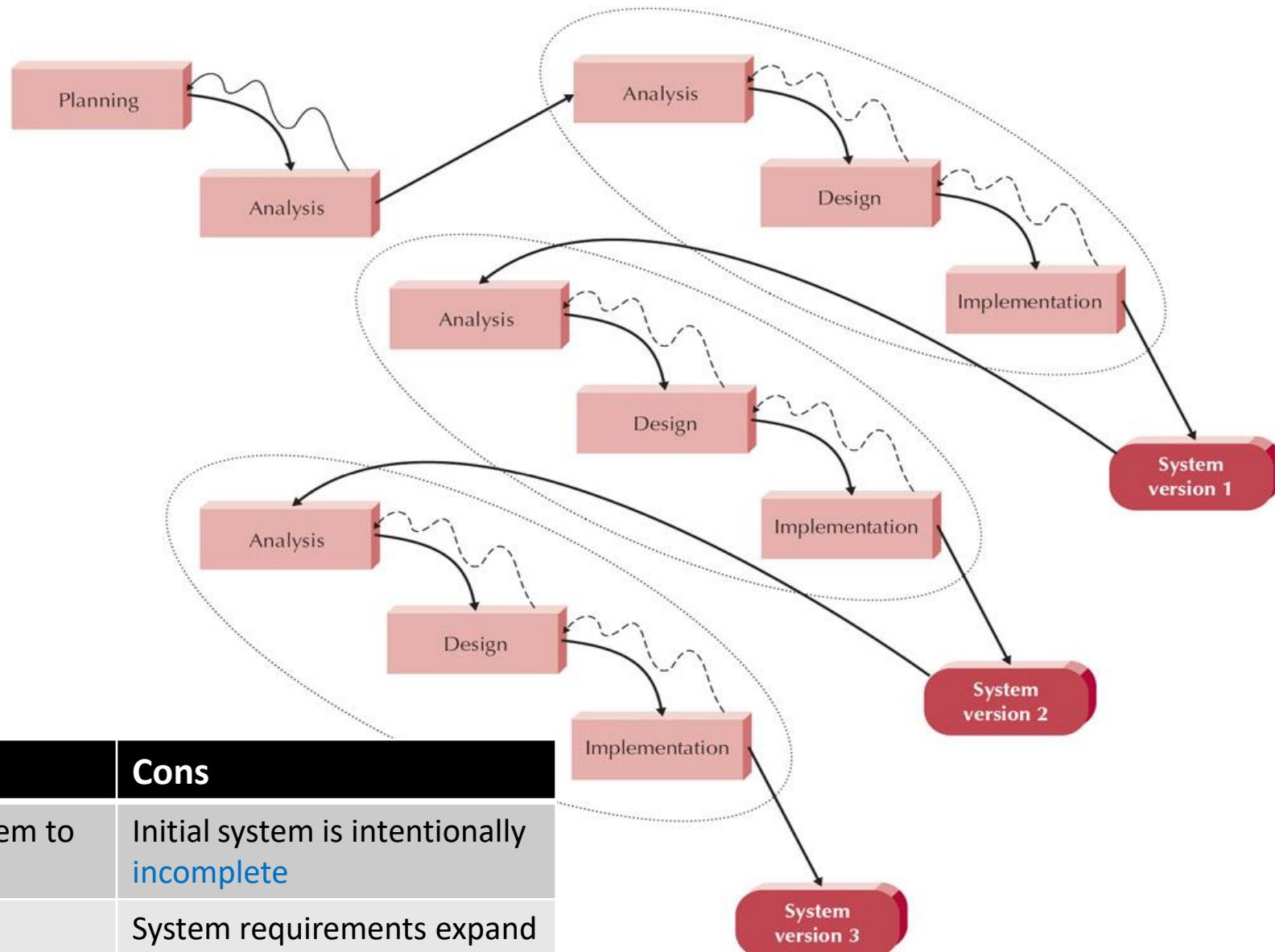
# Parallel Development

Addresses problem of time gap between proposal and delivery:

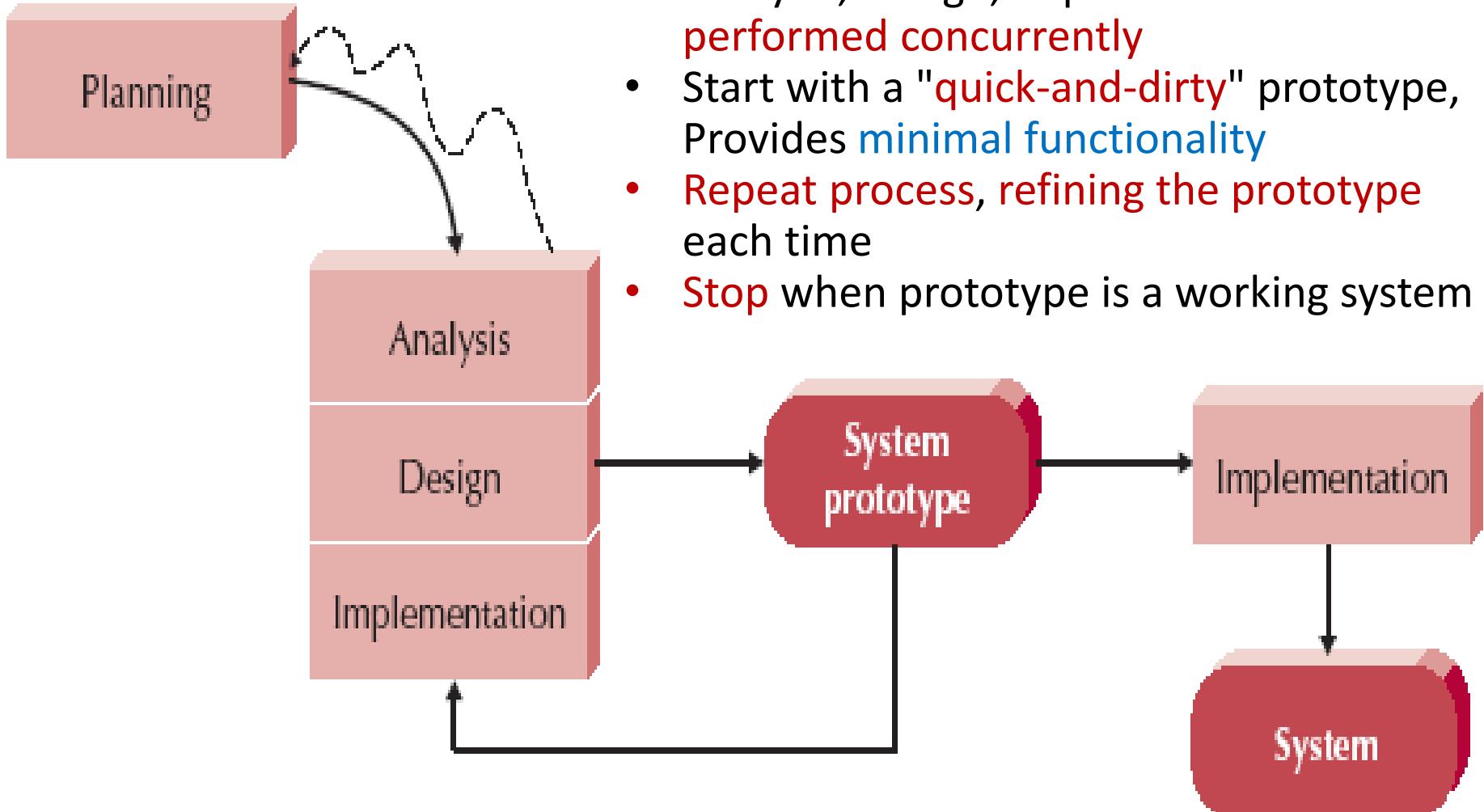
- Breaks project into parallel subproject
- Integrates them at the end



# RAD: Phased Development



# RAD: Prototyping



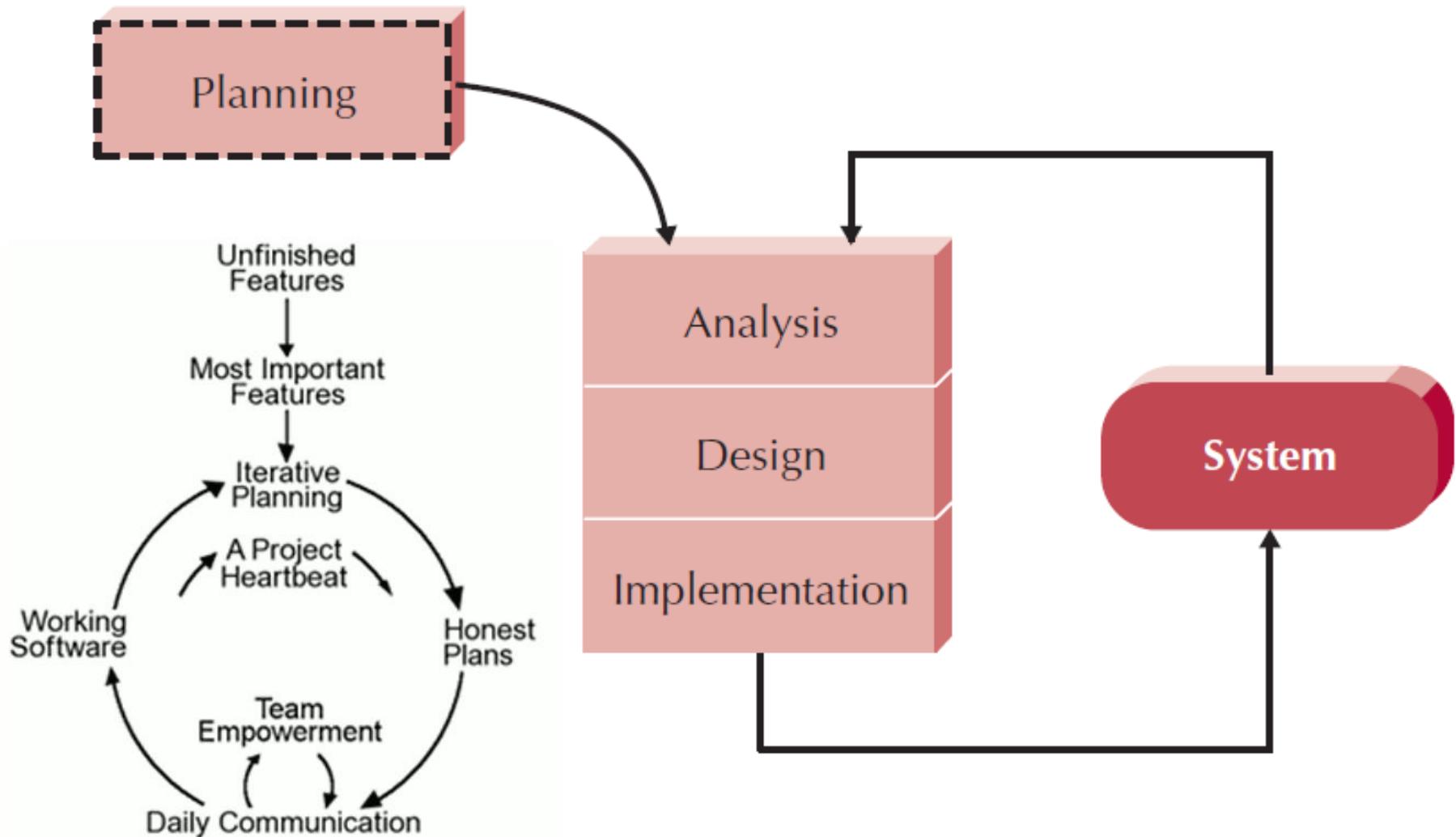
Prototyping significantly **reduces requirement and design errors**, especially for user interfaces (*Boehm's First Law, Endres, 2013*) [**L3**]



# Agile Development

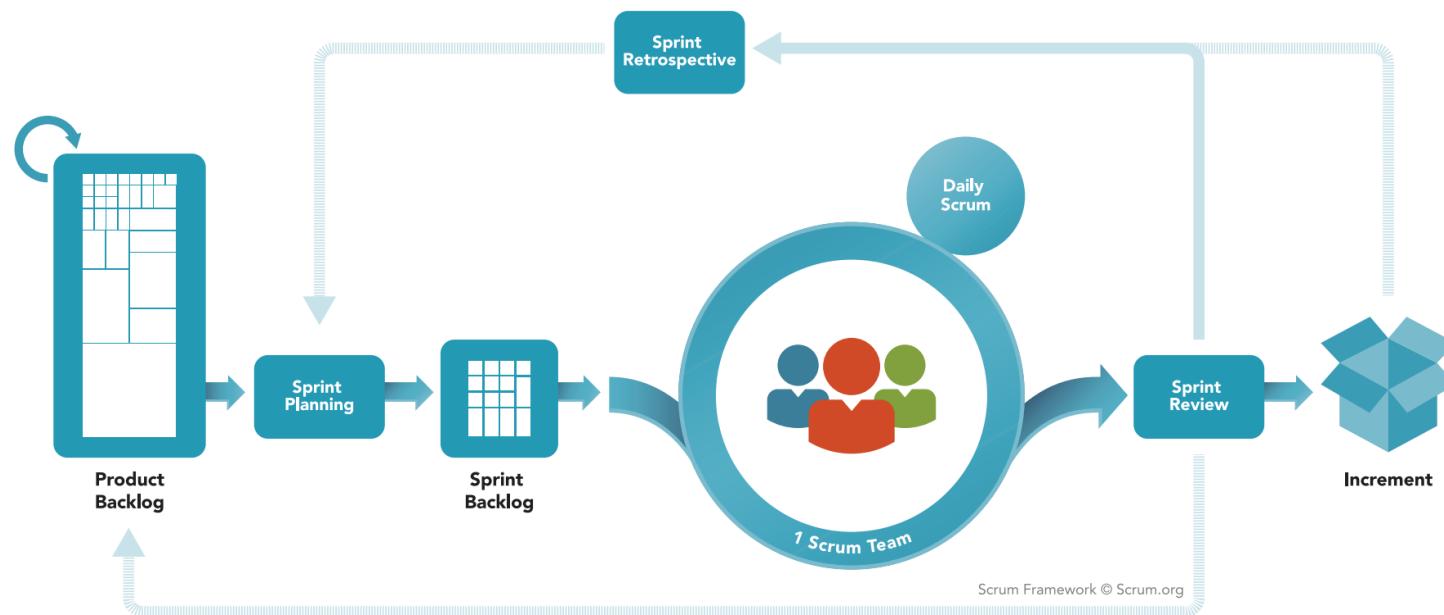
- Menggunakan beberapa aturan yang mudah dipahami dan diikuti
- Mempercepat proses SDLC
  - Mengurangi **pemodelan** dan **dokumentasi**
  - Mengembangkan software dengan **simple** dan **iteratif**
- Contoh Agile Development Practices:
  1. Extreme Programming (XP) (*Kent Beck*)
  2. Scrum (*Ken Schwaber and Jeff Sutherland*)
  3. Lean Development (*Mary Poppendieck and Tom Poppendieck*)
  4. Dynamic Systems Development Model (DSDM) (*Dane Faulkner*)

# Extreme Programming (XP)



# Scrum

- Tim terdiri dari **3-9 pengembang**
- Tujuan dari **Sprint** adalah **menentukan, memprioritaskan, dan memecah** pekerjaan menjadi tugas yang lebih detail
- Tim memiliki karakter **self-organized**, bisa mengelola diri sendiri, memahami tugas, dan bertanggungjawab terhadap deadline
- Setiap Sprint menghasilkan fitur atau **produk yang memiliki nilai** dan menambahkan/memperbaiki fungsi sesuai kebutuhan user/product owner



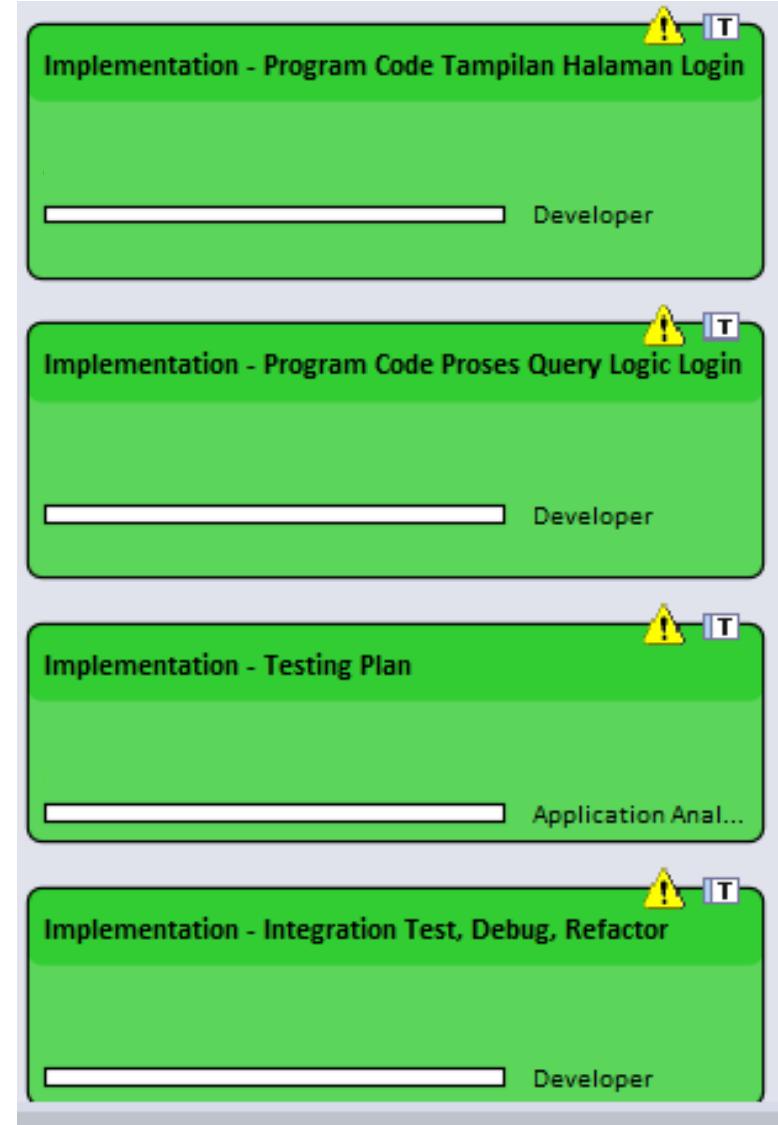
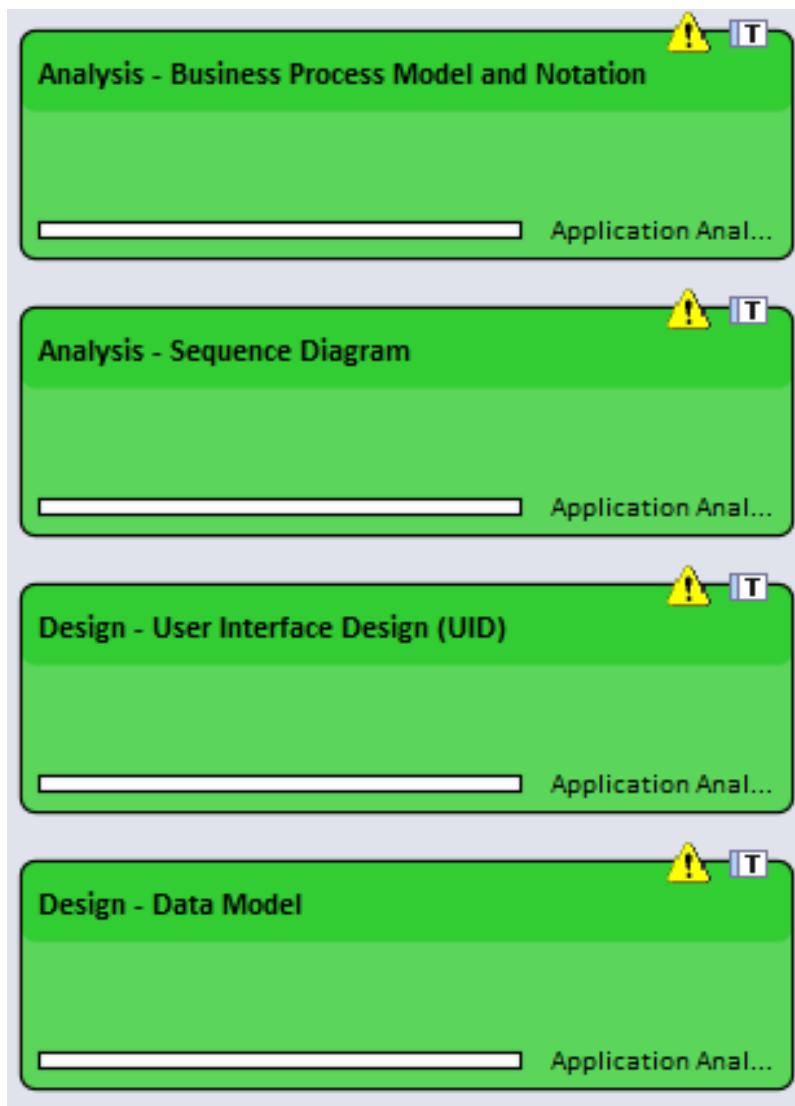
# Strategi Pemilihan Metodologi - Matriks

Ability to Develop Systems	Structured Methodologies		RAD Methodologies			Agile Methodologies	
	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP	SCRUM
With Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent	Excellent
With Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Good	Good
That Are Complex	Good	Good	Good	Poor	Poor	Good	Good
That Are Reliable	Good	Good	Good	Poor	Poor	Excellent	Excellent
With a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent	Excellent
With Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Excellent	Excellent

# Sprint Backlog dengan Task-Tasknya

BACKLOG	TODO	DOING	DONE	
Melakukan Pembelian Lagu  Story Point = 6  Deskripsi PBI: <a href="#">See more</a>	Analysis - Business Process Model and Notation  Analysis - Sequence Diagram  Design - User Interface Design (UID)			<p><b>Sprint Goal</b></p> <p>Pelanggan dapat melakukan pembelian lagu, mengubah data pribadinya dan logout dari sistem MusicPedia</p> <p>Admin dapat mengubah dan menghapus data lagu</p>
Mengubah Data Pribadi  Story Point = 6  Deskripsi PBI: <a href="#">See more</a>	Design - Data Model  Implementation - Program Code Tampilan Halaman Pembelian Lagu			<p><b>Burdon Chart (Sprint 2)</b></p>
Mengubah Data Lagu  Story Point = 6  Deskripsi PBI: <a href="#">See more</a>	Implementation - Program Code Proses Pembelian Lagu  Implementation - Testing Plan			<p><b>Definition of Done (Sprint 2)</b></p> <ul style="list-style-type: none"><li><input type="checkbox"/> Melakukan review terhadap dokumentasi BPMN</li><li><input type="checkbox"/> Melakukan review terhadap dokumentasi Sequence Diagram</li><li><input type="checkbox"/> Dibuatkan Rancangan User Interface Design</li><li><input type="checkbox"/> Kode program diberikan komentar penjelasan pada alur logika/pada setiap function</li><li><input type="checkbox"/> Kode program mengikuti aturan standar dari developer Brainmatics</li><li><input type="checkbox"/> Melakukan Unit Testing</li><li><input type="checkbox"/> Melakukan Integration Testing</li><li><input type="checkbox"/> Melakukan System Testing</li><li><input type="checkbox"/> Melakukan User Acceptance Testing</li><li><input type="checkbox"/> Melakukan Go Live di Production Server</li></ul> <p>Checklist = &lt;memo&gt;</p>
Menghapus Data Lagu  Story Point = 6  Deskripsi PBI: <a href="#">See more</a>	Implementation - Integration Test, Debug, Refactor  Analysis - Sequence Diagram			<p><b>Product Backlog</b></p>
Melakukan Logout  Story Point = 6  Deskripsi PBI: <a href="#">See more</a>	Analysis - Business Process Model and Notation  Design - Data Model			

# Contoh: Task pada PBI Melakukan Login



# MITOS 2

Tidak Mengikuti Panduan yang sudah Dibuat Scrum Guide dan Malah Membuat Improvement yang Akhirnya Gagal



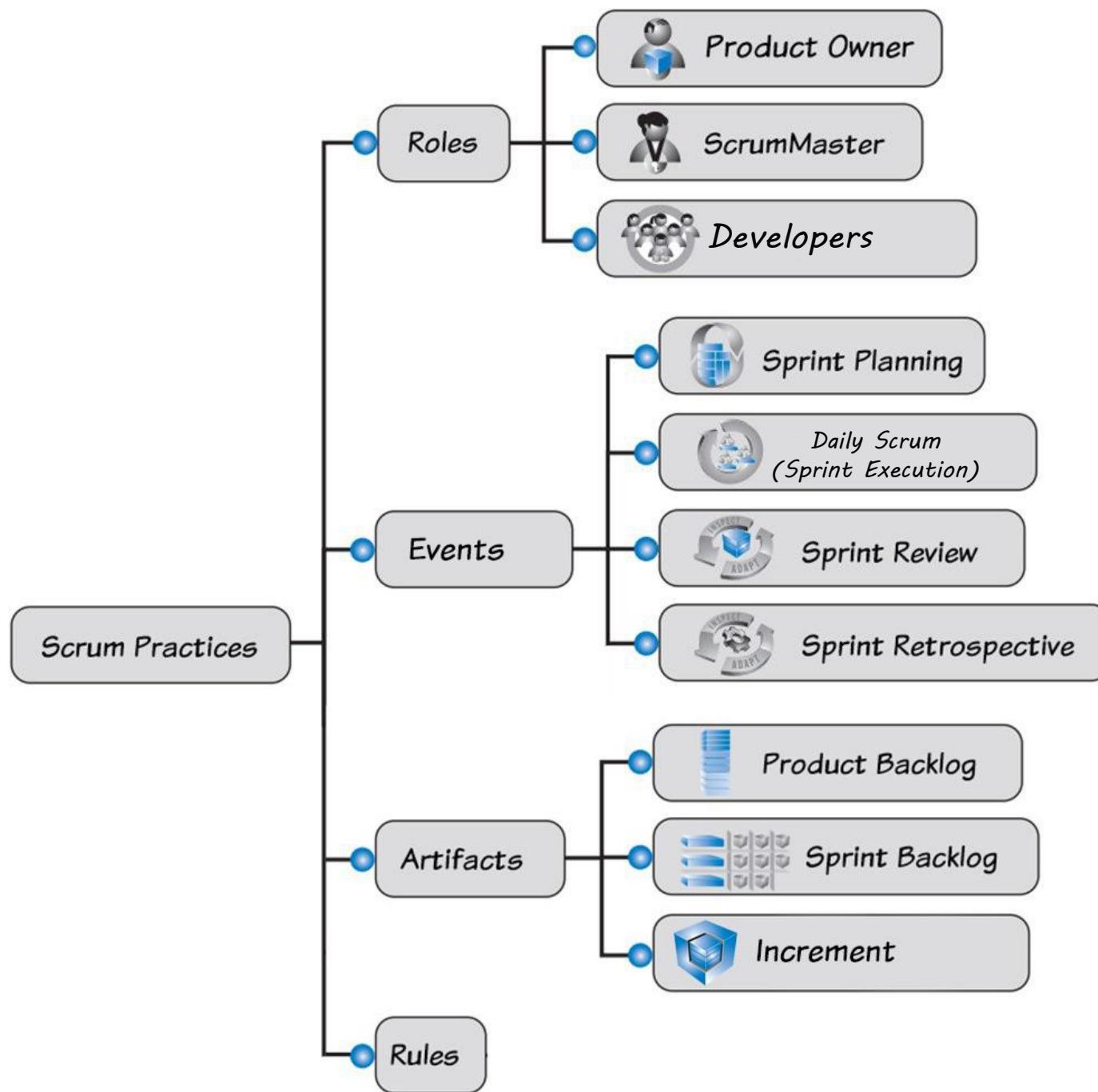
# Purpose of the Scrum Guide

We developed Scrum in the early 1990s. We wrote the first version of the Scrum Guide in 2010 to help people worldwide understand Scrum. We have evolved the Guide since then through small, functional updates. Together, we stand behind it.

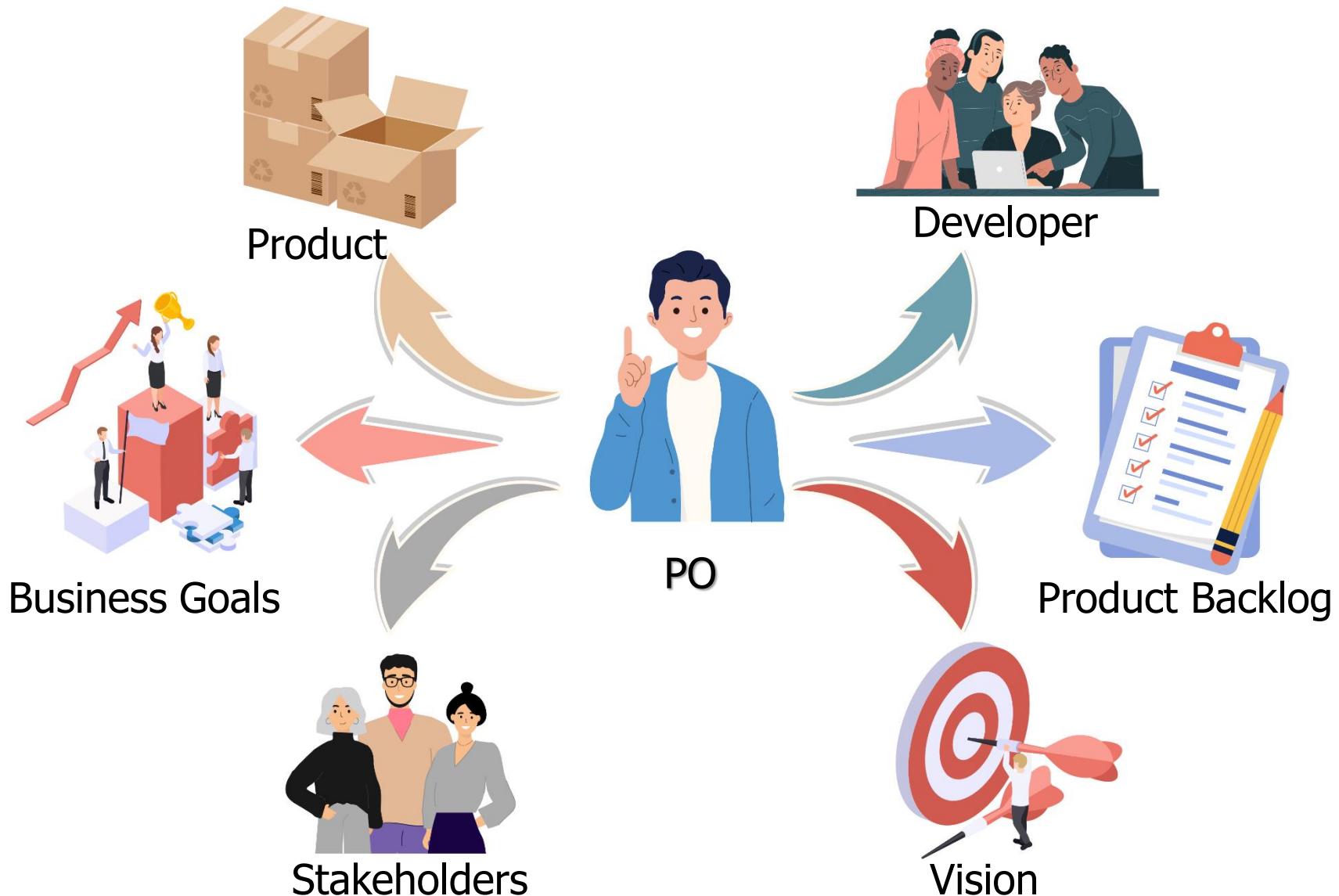
The Scrum Guide contains the definition of Scrum. Each element of the framework serves a specific purpose that is essential to the overall value and results realized with Scrum. Changing the core design or ideas of Scrum, leaving out elements, or not following the rules of Scrum, covers up problems and limits the benefits of Scrum, potentially even rendering it useless.

We follow the growing use of Scrum within an ever-growing complex world. We are humbled to see Scrum being adopted in many domains holding essentially complex work, beyond software product development where Scrum has its roots. As Scrum's use spreads, developers, researchers, analysts, scientists, and other specialists do the work. We use the word "developers" in Scrum not to exclude, but to simplify. If you get value from Scrum, consider yourself included.

As Scrum is being used, patterns, processes, and insights that fit the Scrum framework as described in this document, may be found, applied and devised. Their description is beyond the purpose of the Scrum Guide because they are context sensitive and differ widely between Scrum uses. Such tactics for using within the Scrum framework vary widely and are described elsewhere.



# Product Owner



# Developer



Increment



Developers



Sprint Backlog



Sprint Goal



Definition of Done

# Scrum Master

Servant Leader



Facilitator



Coach

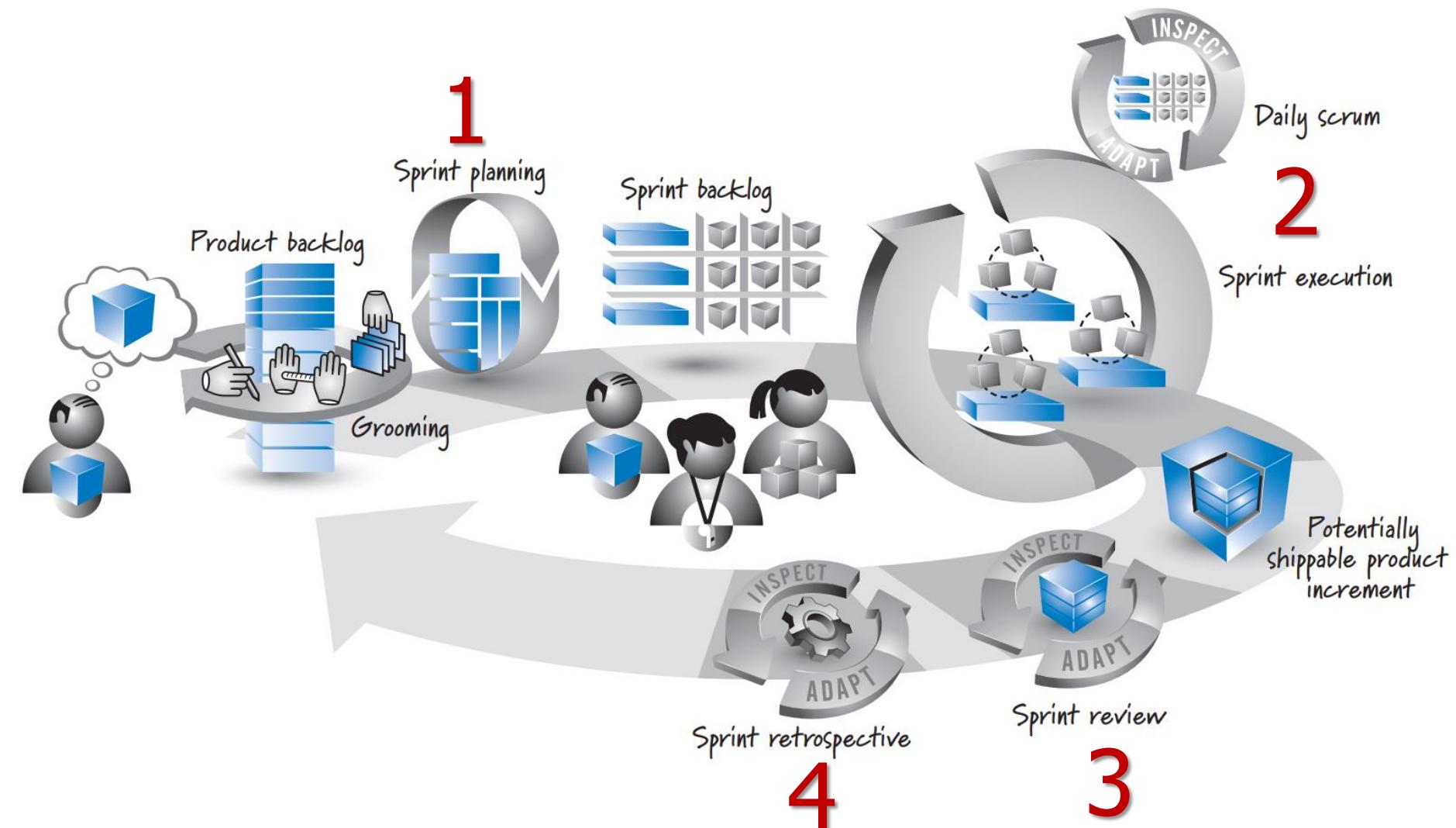


SM

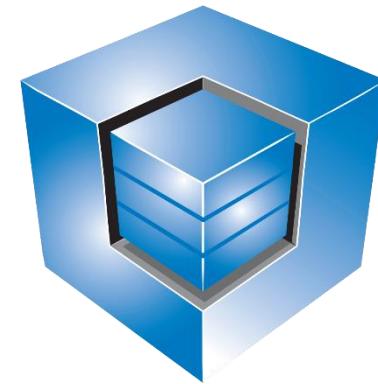
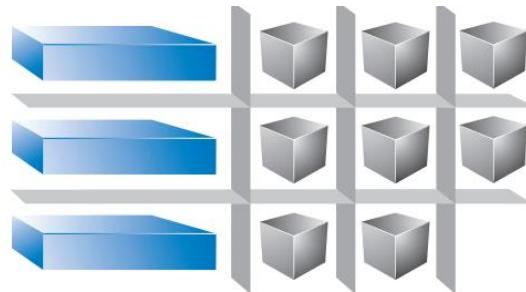
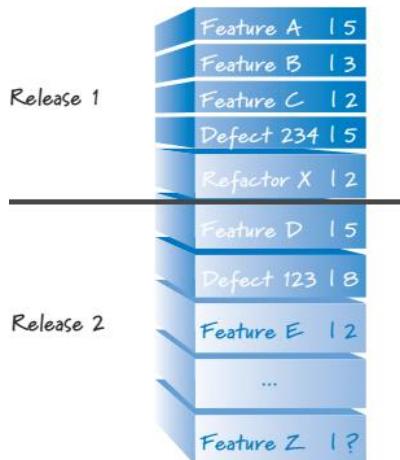
Teacher



# Events



# Artifacts



## 1. Product Backlog

- Dikelola oleh **Product Owner**
- Berisi daftar **fitur, fungsi, kebutuhan, peningkatan, perbaikan** terhadap produk yang akan dirilis
- Memiliki **Product Goal**

## 2. Sprint Backlog

- Dikelola oleh **Developers**
- Berisi daftar **Product Backlog Item** yang terpilih untuk dikerjakan dalam Sprint beserta **Task yang dikerjakan**
- Memiliki **Sprint Goal**

## 3. Increment

- Increment adalah **hasil** dari **Product Backlog Item** yang telah **diselesaikan** dalam Sprint
- Increment adalah **total** nilai bisnis increment dari **seluruh Sprint** yang sebelumnya telah diselesaikan
- Memiliki **Definition of Done**

# Scrum Definition

Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.

In a nutshell, Scrum requires a Scrum Master to foster an environment where:

1. A Product Owner orders the work for a complex problem into a Product Backlog.
2. The Scrum Team turns a selection of the work into an Increment of value during a Sprint.
3. The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.
4. *Repeat*

Scrum is simple. Try it as is and determine if its philosophy, theory, and structure help to achieve goals and create value. The Scrum framework is purposefully incomplete, only defining the parts required to implement Scrum theory. Scrum is built upon by the collective intelligence of the people using it. Rather than provide people with detailed instructions, the rules of Scrum guide their relationships and interactions.

Various processes, techniques and methods can be employed within the framework. Scrum wraps around existing practices or renders them unnecessary. Scrum makes visible the relative efficacy of current management, environment, and work techniques, so that improvements can be made.

# Product Backlog

- Product Backlog adalah **daftar urutan** seluruh **fitur, fungsi, kebutuhan, peningkatan** dan **perbaikan** yang **diperlukan** sebuah produk
- Product Backlog **berevolusi** seiring dengan **perkembangan** produk
- Product Backlog memiliki **Product Goal** untuk mendefinisikan tujuan dari produk.
- Product Owner **bertanggungjawab** terhadap Product Backlog  
*(Scrum Guide, 2020)*

PRODUCT BACKLOG		
PRODUCT BACKLOG ITEM (PBI)		
LOW IMPORTANCE	MEDIUM	HIGH IMPORTANCE
Item 1	Item 2	Item 3
Item 4	Item 5	Item 6
Item 7	Item 8	Item 9
Item 10	Item 11	Item 12
Item 13	Item 14	Item 15
Item 16	Item 17	Item 18
Item 19	Item 20	Item 21
Item 22	Item 23	Item 24
Item 25	Item 26	Item 27
Item 28	Item 29	Item 30
Item 31	Item 32	Item 33
Item 34	Item 35	Item 36
Item 37	Item 38	Item 39
Item 40	Item 41	Item 42
Item 43	Item 44	Item 45
Item 46	Item 47	Item 48
Item 49	Item 50	Item 51
Item 52	Item 53	Item 54
Item 55	Item 56	Item 57
Item 58	Item 59	Item 60
Item 61	Item 62	Item 63
Item 64	Item 65	Item 66
Item 67	Item 68	Item 69
Item 70	Item 71	Item 72
Item 73	Item 74	Item 75
Item 76	Item 77	Item 78
Item 79	Item 80	Item 81
Item 82	Item 83	Item 84
Item 85	Item 86	Item 87
Item 88	Item 89	Item 90
Item 91	Item 92	Item 93
Item 94	Item 95	Item 96
Item 97	Item 98	Item 99
Item 100	Item 101	Item 102



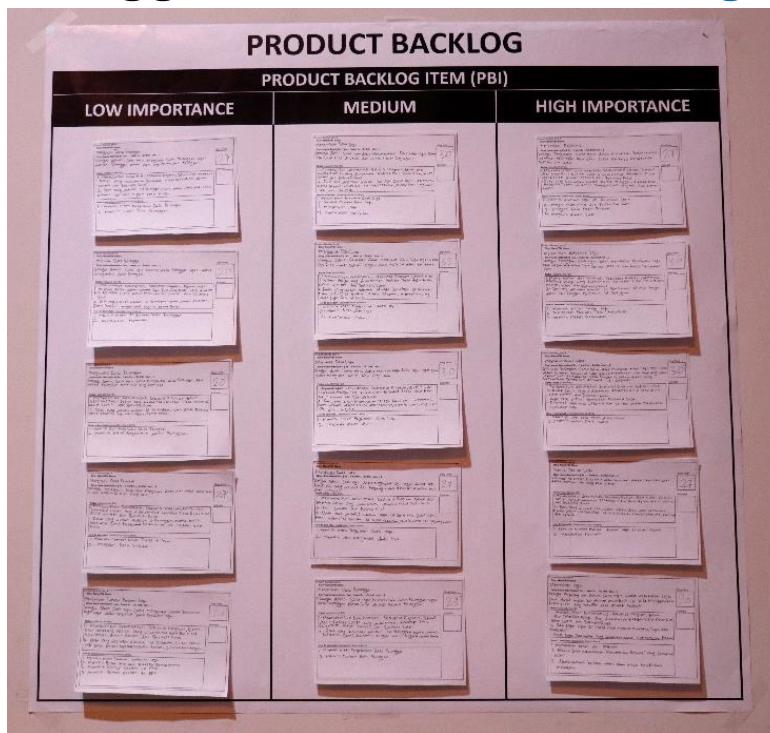
# Product Backlog Items Prioritized

- PBI yang memiliki prioritas **tinggi** diletakkan di kategori **HIGH IMPORTANCE** dan dikerjakan **paling awal**
- PBI yang memiliki prioritas **sedang** diletakkan di kategori **MEDIUM** dan dikerjakan **setelah kategori HIGH IMPORTANCE**
- PBI yang memiliki prioritas **rendah** diletakkan di kategori **LOW IMPORTANCE** dan dikerjakan **setelah kategori MEDIUM**

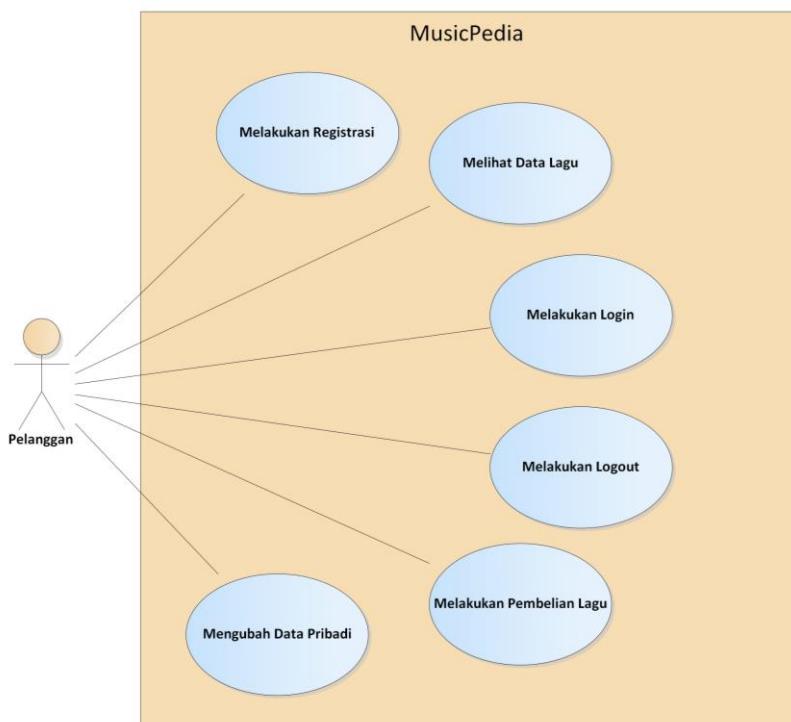


# Product Backlog Item dengan Use Case Diagram

- Belum ada **standar** yang mengatur **format** dari Product Backlog Item, beberapa orang menggambarkan dengan **User Stories** (*Kenneth S. Rubin, 2012*)
- Product Backlog Item **dapat digambarkan** dengan menggunakan **Use Case Diagram** (*Kenneth S. Rubin, 2012*)



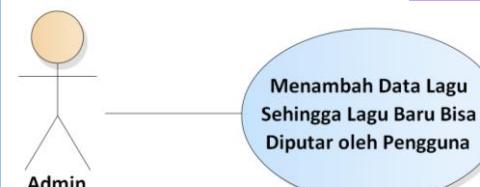
Product Backlog Item



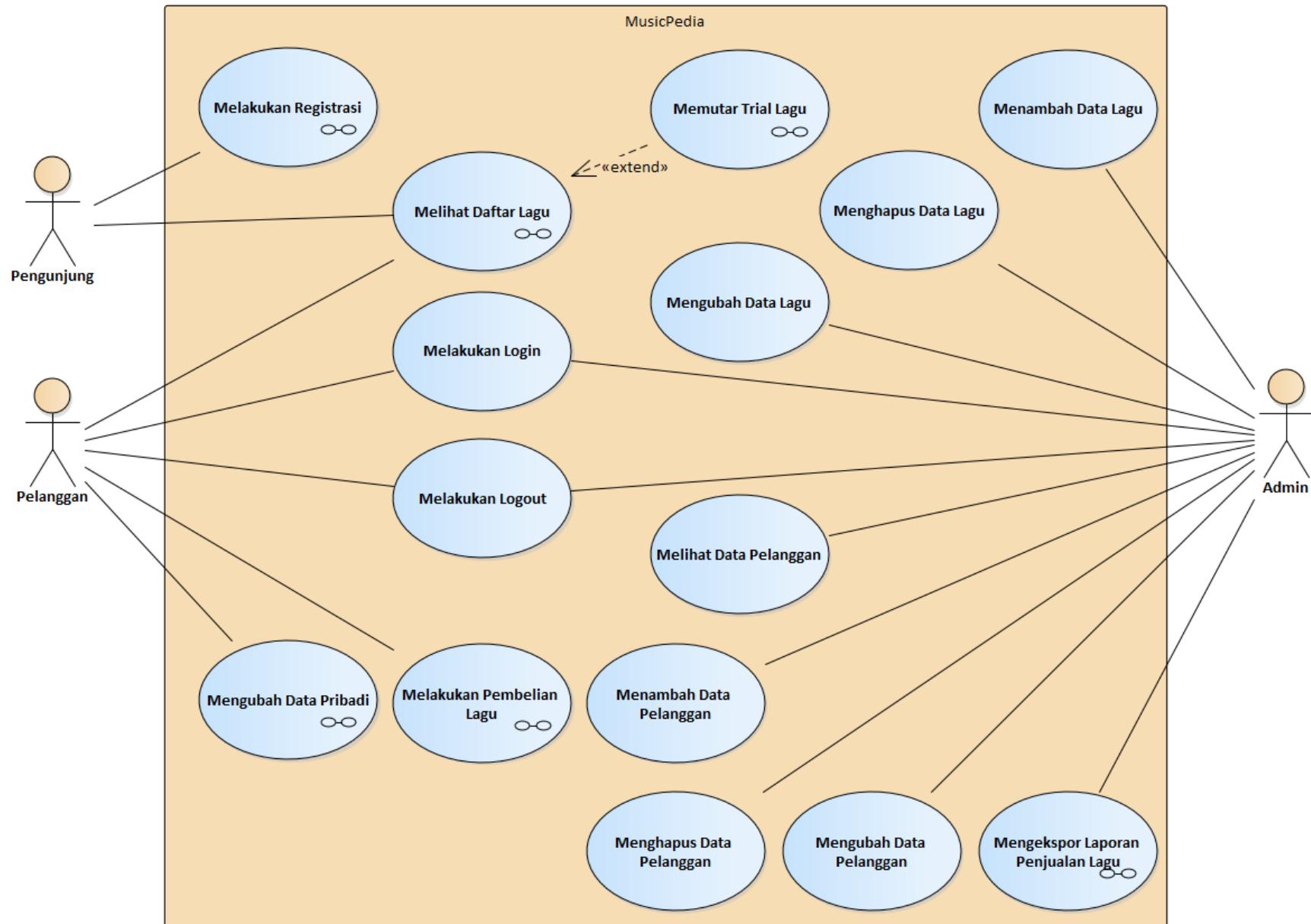
Use Case Diagram

# Product Backlog Item Template

## Product Backlog Item (PBI)

Nama PBI	Poin PBI
Nama PBI dapat berupa <i>feature, defect, technical improvement, knowledge acquisition</i>	
Deskripsi PBI, dapat diisi dengan <i>As....</i> <i>I Want to...</i> <i>So that I can...</i>	Diisi berdasarkan penghitungan <i>story points</i>
 <p>Menambah Data Lagu Sehingga Lagu Baru Bisa Diputar oleh Pengguna</p>	
Catatan untuk PBI, bisa terkait catatan tentang dokumentasi, atau kebutuhan <i>environment</i> dalam rangka mengerjakan PBI	Estimasi PBI (Hari)
Catatan cara penggunaan dan pengujian fitur pada sistem, dapat diisi dengan Activity Diagram/BPMN dan <i>test plan</i>	Estimasi dari peng kerjaan PBI dalam satuan hari

# Contoh: Use Case Diagram MusicPedia



# Product Backlog Grooming

1. Mengidentifikasi Product Backlog Item
2. Menentukan prioritas Product Backlog Item



Product Backlog Item (PBI)		Poin PBI
Nama PBI Menjejakspor Laporan Penjualan Lagu		
Deskripsi PBI Sebagai admin, saya ingin dapat menjejakspor laporan penjualan lagu agar dapat melihat hasil penjualan lagu		
Catatan PBI <ul style="list-style-type: none"><li>- Membutuhkan dokumentasi sequence diagram, BPMN, User Interface Design yang didalamnya terdapat field description</li><li>- Data yang ditemui adalah nama pelanggan, alamat email, password, jenis kelamin, Tanggal lahir, Nama ayah dan nomor ponsel</li></ul>		Estimasi PBI (Hari)
Cara Demo dan Pengujian PBI <ol style="list-style-type: none"><li>1. Memilih menu laporan penjualan lagu</li><li>2. Memilih bolan yang akan diakses penjualannya,</li><li>3. Memilih tombol EXPOR TO EXCEL</li><li>4. memilih tombol EXPOR TO PDF</li></ol>		

# Pelaksanaan Sprint Planning

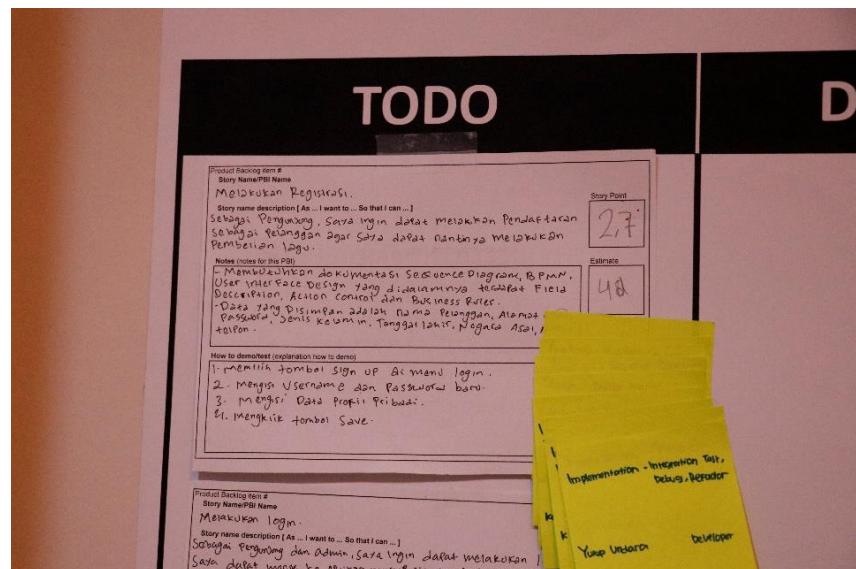
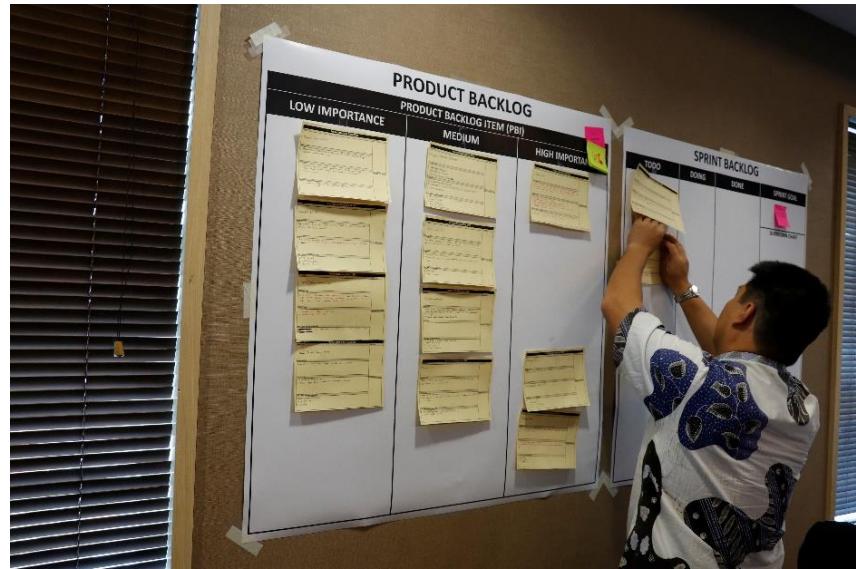
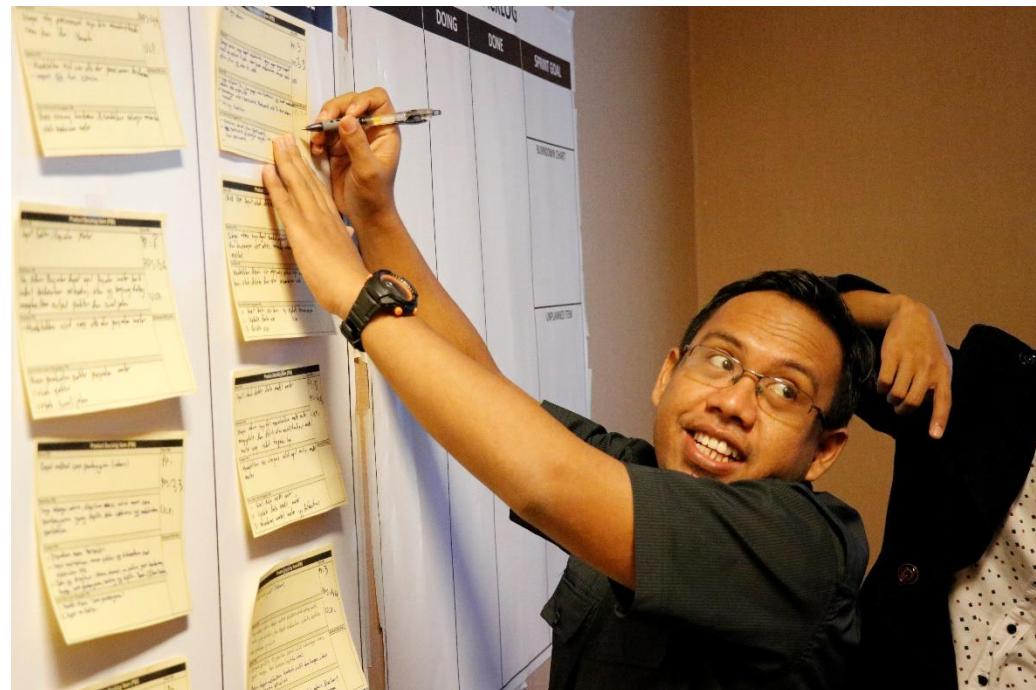


# Contoh: Task pada Sprint Backlog



# Metode Estimasi Product Backlog Item

1. Planning Poker (PP)
2. Points Per Story (PPS)
3. Use Case Points (UCP)
4. Velocity Calculations (VC)



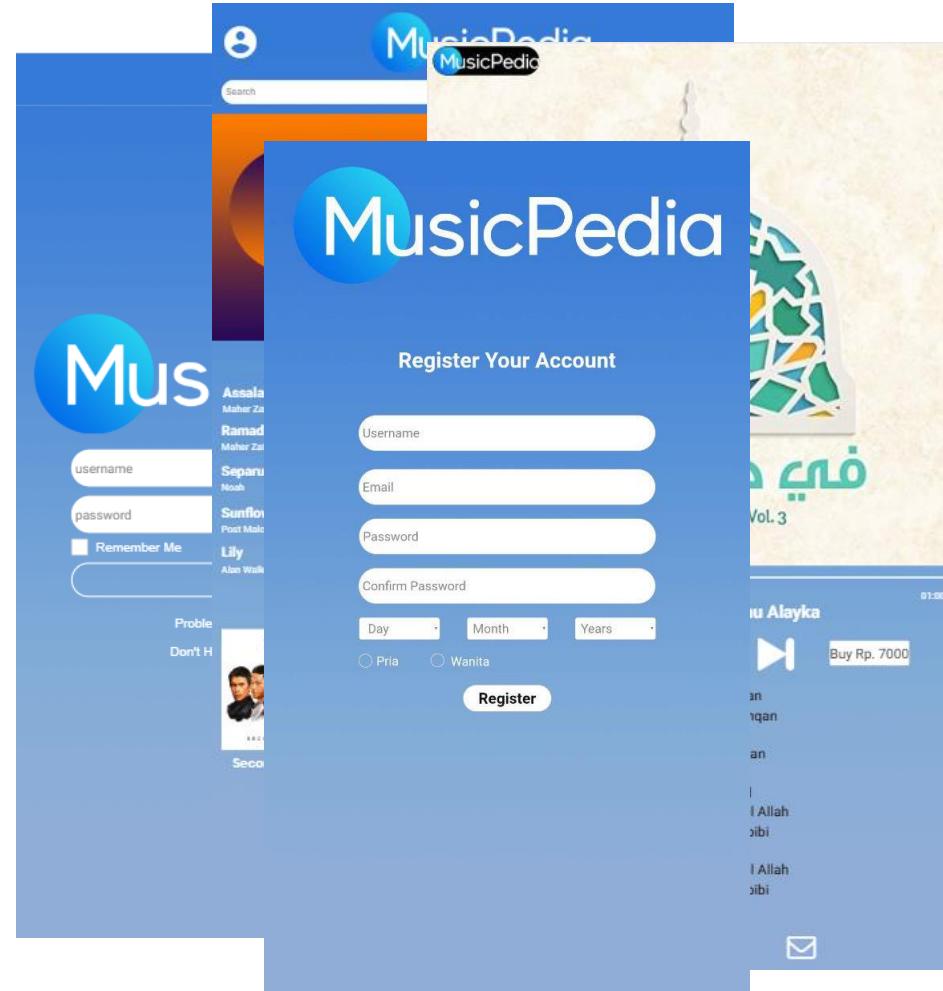
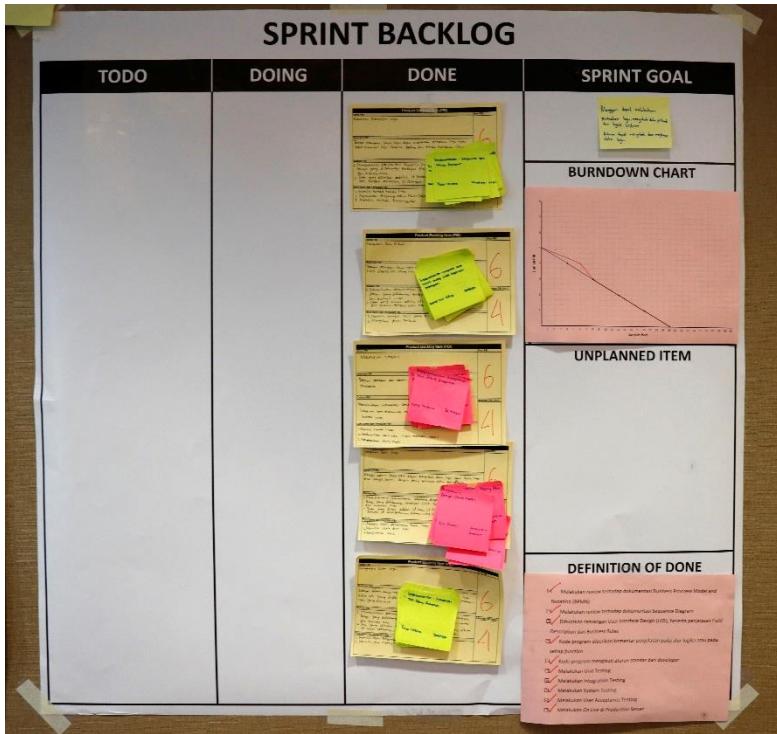
# Daily Scrum

- Daily Scrum dilakukan **setiap hari** selama Sprint berlangsung
- Isi diskusi Daily Scrum:
  1. Aktivitas pada **aktivitas hari kemarin**
  2. **Kendala** pada aktivitas hari kemarin
  3. Rencana aktivitas hari ini, untuk membantu tim mencapai Sprint Goal



# Output Sprint Execution

- **Potentially Shippable Product Increment** adalah produk yang dihasilkan pada saat **Sprint Execution** yang dilakukan oleh Scrum Team



# Proses Sprint Review

- **Overview:** Melakukan pemeriksaan dan penyesuaian terhadap Product Baklog Item
- **Demonstrate:** Demonstrasi sebenarnya hanyalah cara yang sangat efisien untuk memulai diskusi
- **Discuss:** Fokus diskusi utama dalam Sprint Review adalah melakukan diskusi untuk menyempurnakan produk yang dihasilkan
- **Adapt:** Ketika disepakati untuk dilakukan perubahan, maka Product Backlog akan diperbarui



# Proses Sprint Retrospective



# Output Sprint Retrospective

- *Improvement Actions*: Merencanakan **aksi improvisasi** untuk melakukan **perbaikan kinerja** dari sprint sebelumnya agar dapat **meningkatkan kinerja** yang lebih baik pada sprint berikutnya
- *Insight Backlog*: Daftar rencana perbaikan untuk meningkatkan kualitas produk backlog pada sprint berikutnya

Good	Could Have Been Better	Improvements
Burndown Chart bentuknya Sudah bagus	Product Backlog ditambahkan Kolom untuk nama aplikasi	Disediakan coffee break Saat sprint planning berikutnya karena pada sprint planning sebelumnya peserta yg mengikuti sprint planning tidak diberikan minum

# MITOS 3

Masih Memikirkan **Dokumentasi Pengembangan**, dan Kurang Fokus ke Pengembangan Produk



# Major Software Development Methodologies

## 1. Structured Design

(**Prescriptive**) (1967- )

- Waterfall method
- Parallel development



More  
Prescriptive/  
Documentation

## 2. Rapid Application Development

(**Iterative**) (1985-)

- Phased Development
- Prototyping



More  
Adaptive/  
Communication

## 3. Agile Development

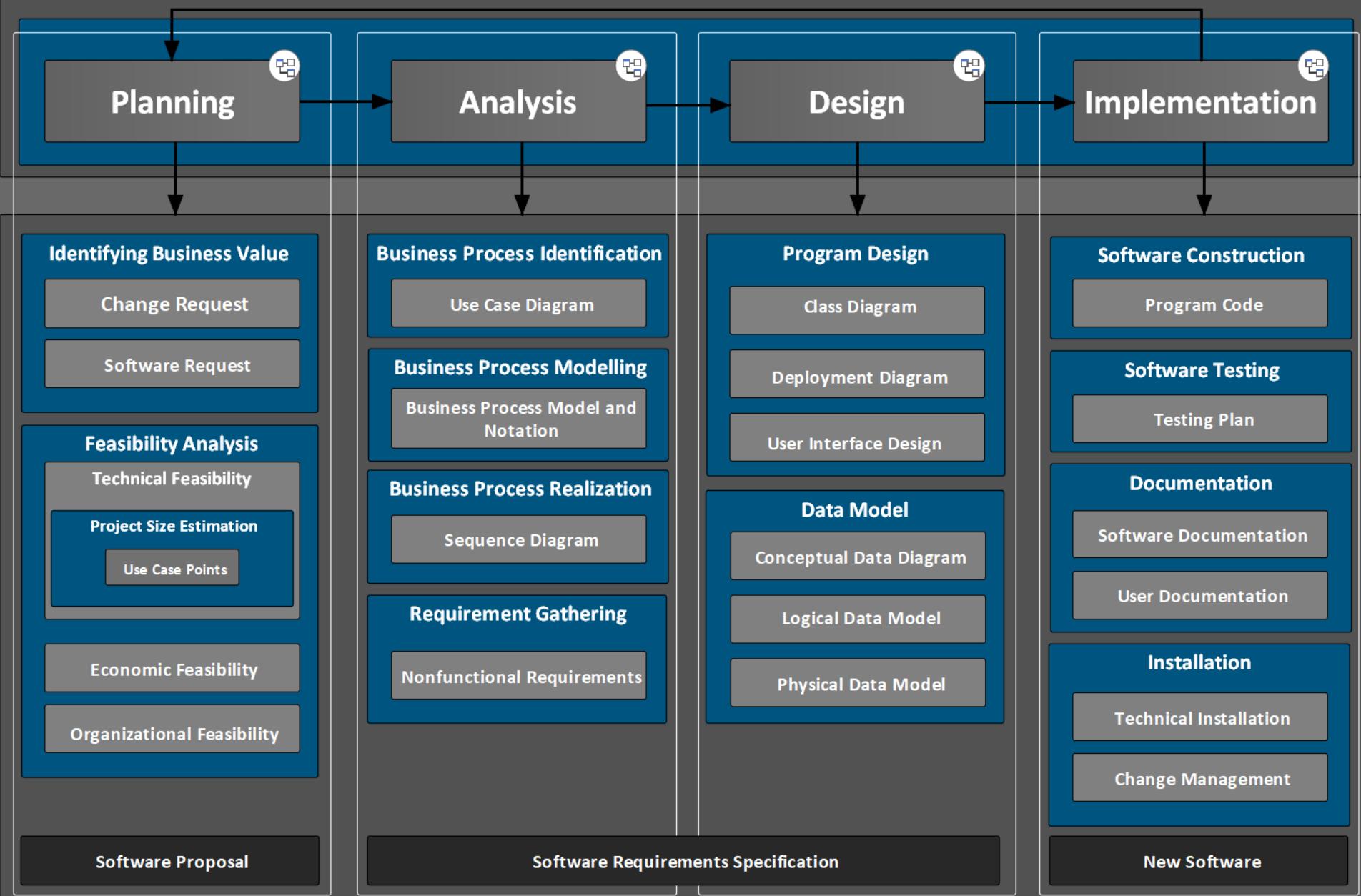
(**Adaptive**) (1995-)

- Extreme Programming (XP)
- Scrum

*Compiled from (Dennis, Wixom and Tegarden, 2016)*

# Application Development Governance

## Software Development Life Cycle



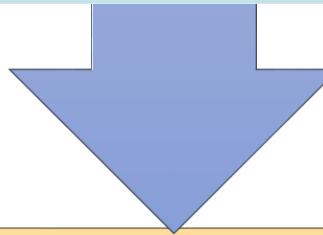
# Planning

## System Request (Business Value Identification)

*Lower Cost*

*Increase  
Productivity*

*Increase Profit*

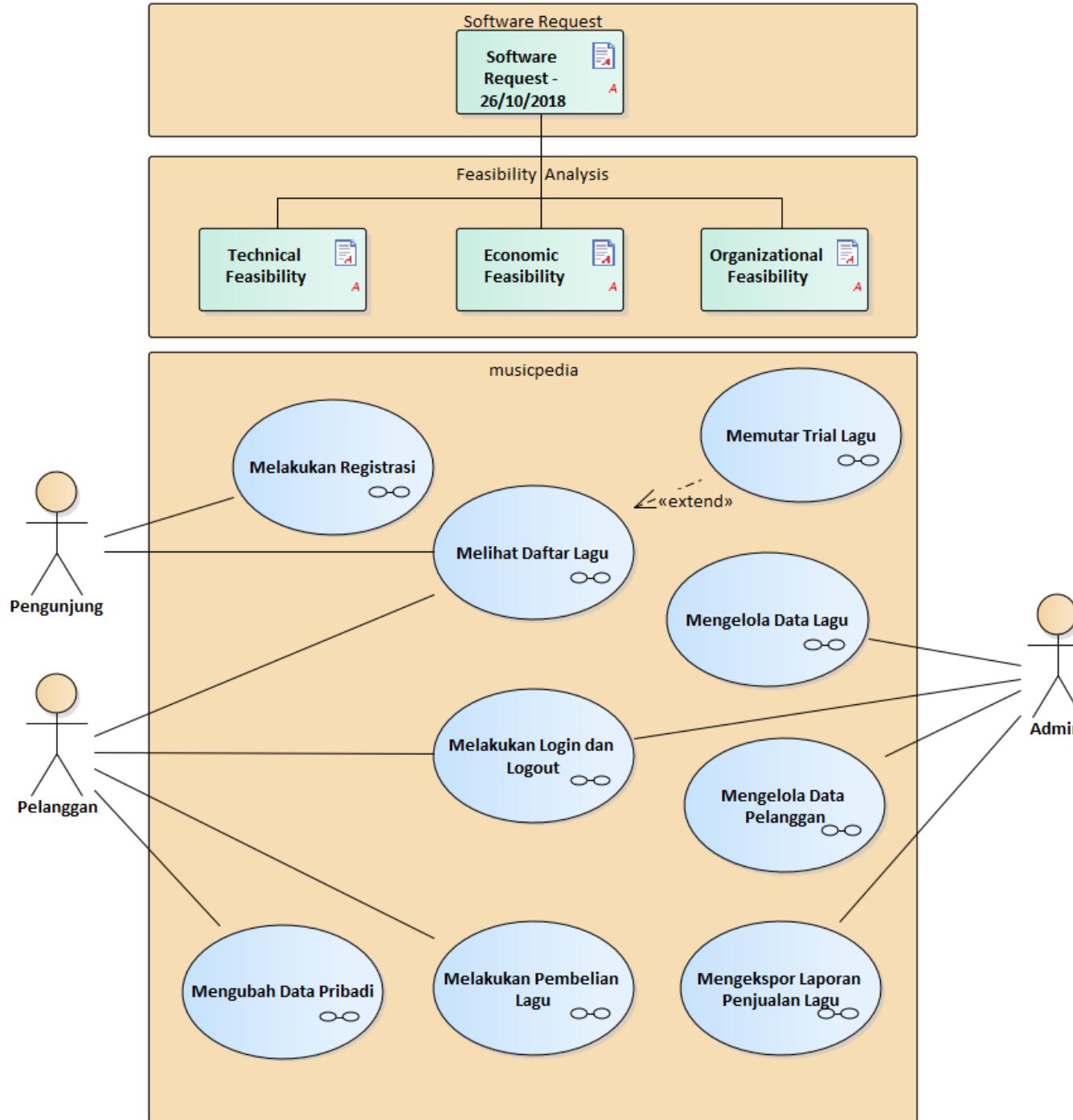


## Feasibility Analysis

*Technical  
(Capabilities)*

*Economic  
(ROI, BEP)*

*Organizational  
(Goals, Core Business)*



# System Request: Sistem Penjualan Musik Online

<b>Project Sponsor:</b>	Margaret Mooney, Vice President of Marketing
<b>Business Needs:</b>	Project ini dibangun untuk: <ol style="list-style-type: none"><li>1. Mendapatkan pelanggan baru lewat Internet</li><li>2. Meningkatkan efisiensi penanganan masalah pelanggan melalui internet</li></ol>
<b>Business Requirements:</b>	Sistem yang mendukung penjualan musik secara online. Fitur-fitur yang harus ada: <ol style="list-style-type: none"><li>1. Fitur Pencarian Produk</li><li>2. Fitur Pencarian Toko yang Menyediakan Stok Produk</li><li>3. Fitur Pemesanan Produk Melalui Toko yang Menyediakan</li><li>4. Fitur Pembayaran dengan Berbagai Pilihan Pembayaran</li></ol>
<b>Business Value:</b>	
<i>Intangible Value:</i>	<ul style="list-style-type: none"><li>▪ Meningkatkan kenyamanan dan <b>kepuasan pelanggan</b></li><li>▪ Meningkatkan <b>brand recognition</b> tentang perusahaan di dunia Internet</li></ul>
<i>Tangible Value:</i>	<ol style="list-style-type: none"><li>1. Meningkatkan penjualan dari pelanggan baru lewat Internet:<ul style="list-style-type: none"><li>• Rp 400 juta <b>peningkatan penjualan</b> dari pelanggan baru dan Rp 600 juta dari pelanggan lama</li></ul></li><li>2. Mengurangi biaya operasional untuk menangani komplain dari pelanggan<ul style="list-style-type: none"><li>• Rp 100 juta <b>pengurangan tahunan biaya telepon</b> untuk menangani pelanggan</li></ul></li></ol>

# Software Request

## musicpedia

<b>Date</b>	26 Oktober 2018			
<b>Description</b>	Musicpedia adalah aplikasi layanan download musik & audio dimana saja dan kapan saja, menawarkan akses lengkap ke jutaan lagu dari semua artis papan atas di industri musik barat maupun musik lokal, mendengarkan musik baru dan top musik dunia dalam bentuk audio mp3 secara offline			
<b>Project Sponsor</b>	Wahyu Utomo, VP Business Development, PT Musika Indonesia			
<b>Business Need</b>	1. Tidak Setuju	2. Ragu-Ragu	3. Setuju	4. Sangat Setuju
Aplikasi yang dikembangkan mampu meningkatkan pendapatan perusahaan?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Aplikasi yang dikembangkan mampu mengurangi biaya operasional perusahaan?	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Aplikasi yang dikembangkan mampu meningkatkan produktifitas kerja pegawai?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aplikasi yang dikembangkan mampu meningkatkan nilai tambah perusahaan yang bersifat intangible?	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>Business Value</b>	<p>Intangible Value:</p> <ul style="list-style-type: none"> <li>a. Meningkatkan brand recognition perusahaan di dunia internet</li> <li>b. Meningkatkan produktivitas kerja pegawai dan mengurangi kuantitas pegawai</li> </ul> <p>Tangible Value:</p> <ul style="list-style-type: none"> <li>a. Mengurangi biaya operasional perusahaan: <ul style="list-style-type: none"> <li>- Sewa ruangan: Rp120.000.000,-</li> <li>- Biaya komunikasi: Rp6.000.000,-</li> </ul> </li> <li>b. Meningkatkan penjualan musik: Rp400.000.000,-</li> </ul>			

# Technical Feasibility

## musicpedia

Date: 26 Oktober 2018

Penjelasan isian	1. Sangat Kurang	2. Kurang	3. Baik	4. Sangat Baik
Kefamiliaran dengan Aplikasi	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pengguna familiar terhadap pengoperasian aplikasi ini.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pengembang familiar terhadap pengembangan aplikasi ini.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Kefamiliaran dengan Teknologi	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pengguna familiar dengan teknologi pendukung aplikasi.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pengembang familiar mengembangkan aplikasi dengan platform, bahasa pemrograman dan tool IDE yang dipilih.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Ukuran Proyek				
Jumlah pengembang yang dibutuhkan.	7 Man/Month			
Waktu yang dibutuhkan dalam mengembangkan aplikasi ini.	6 Month			
Kompatibilitas	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Kebutuhan pengguna terhadap kompatibilitas aplikasi untuk terintegrasi dengan aplikasi lain.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Kompatibilitas aplikasi terhadap teknologi yang ada pada organisasi.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Secara analisis kelayakan teknis, apakah aplikasi layak dikembangkan sesuai kriteria di atas?	<input checked="" type="checkbox"/>	Layak	<input type="checkbox"/>	Tidak Layak

# Technical Feasibility

## Use Case Points

### Tahap 1 - Menghitung Person Hours (PH)

Use Case Points (UCP)	Person Hours Multiplier (PHM)	Person Hours (PH)
51	20	1020
51	28	1428

### Tahap 2 - Menghitung Person Month (PM)

PHM	Person Hours (PH)	Lama Bekerja Perhari	Jumlah Bekerja Sebulan	Person Months (PM)
20	1020	8	22	5.80
	1020	10	26	3.92
28	1428	8	22	8.11
	1428	10	26	5.49

### Tahap 3 - Menghitung Time (Month)

PHM	Formula Penghitung Waktu	Jumlah Bekerja Sebulan	Waktu dalam Bulan (M)
20	$3 * PM^{(1/3)}$	22	5.39
		26	4.73
		22	6.03
		26	5.29

# Economic Feasibility

## Cost-Benefit Analysis

Tahun	2019	2020	2021	2022
Peningkatan Pendapatan Penjualan Lagu		400,000,000	400,000,000	400,000,000
Pengurangan Biaya Sewa Ruangan		120,000,000	120,000,000	120,000,000
Pengurangan Biaya Komunikasi		6,000,000	6,000,000	6,000,000
<b>Total Benefits</b>	<b>0</b>	<b>526,000,000</b>	<b>526,000,000</b>	<b>526,000,000</b>
<b>PV of Benefits</b>	<b>0</b>	<b>468,138,127</b>	<b>441,639,743</b>	<b>416,641,267</b>
<b>PV of All Benefits</b>	<b>0</b>	<b>468,138,127</b>	<b>909,777,870</b>	<b>884,779,394</b>
Honor Tim (Analysis, Design and Implementation)	250,000,000	120,000,000	120,000,000	120,000,000
<b>Total Development Costs</b>	<b>250,000,000</b>	<b>120,000,000</b>	<b>120,000,000</b>	<b>120,000,000</b>
Honor Pengelola Web	72,000,000	72,000,000	72,000,000	72,000,000
Biaya Lisensi Software	10,000,000	10,000,000	10,000,000	10,000,000
Hardware upgrades	50,000,000	50,000,000	50,000,000	50,000,000
Biaya Komunikasi	1,000,000	1,000,000	1,000,000	1,000,000
Biaya Marketing	50,000,000	50,000,000	50,000,000	50,000,000
<b>Total Operational Costs</b>	<b>183,000,000</b>	<b>183,000,000</b>	<b>183,000,000</b>	<b>183,000,000</b>
<b>Total Costs</b>	<b>433,000,000</b>	<b>303,000,000</b>	<b>303,000,000</b>	<b>303,000,000</b>
<b>PV of Costs</b>	<b>408,490,566</b>	<b>269,668,921</b>	<b>153,650,329</b>	<b>144,953,140</b>
<b>PV of all Costs</b>	<b>408,490,566</b>	<b>678,159,487</b>	<b>831,809,816</b>	<b>976,762,957</b>
<b>Total Project Costs Less Benefits</b>	<b>-433,000,000</b>	<b>223,000,000</b>	<b>223,000,000</b>	<b>223,000,000</b>
<b>Yearly NPV</b>	<b>-408,490,566</b>	<b>198,469,206</b>	<b>187,235,100</b>	<b>176,636,887</b>
<b>Cumulative NPV</b>	<b>-408,490,566</b>	<b>-210,021,360</b>	<b>-22,786,260</b>	<b>153,850,627</b>
<b>Return on Investment (ROI)</b>	<b>-100.00%</b>	<b>-0.309693168</b>	<b>-0.027393593</b>	<b>0.15751071</b>
<b>Break-even Point (BEP)</b>				<b>3.129000574</b>

# Organizational Feasibility

musicpedia		
Date	26 Oktober 2018	
<b>Anggota Tim</b>		
User/Product Owner	Wahyu Utomo	
Project Manager	Haris Dermawan	
System Analyst	Risa Dhani Horasman Purba	
Business Analyst	Mulyana	
Programmer	Achmad Fatkarrofiqi	
Tester	Januar Sapareza	
<b>Apakah aplikasi ini mendukung visi dan misi organisasi?</b>		
Ya		
<b>Apakah aplikasi ini sesuai dengan tugas, fungsi dan KPI unit kerja anda?</b>		
Ya		
<b>Apakah aplikasi ini selaras dengan proses bisnis unit kerja anda?</b>		
Ya		
Secara analisis kelayakan organisasi, apakah aplikasi layak dikembangkan sesuai kriteria di atas?	<input checked="" type="checkbox"/> Layak	<input type="checkbox"/> Tidak Layak

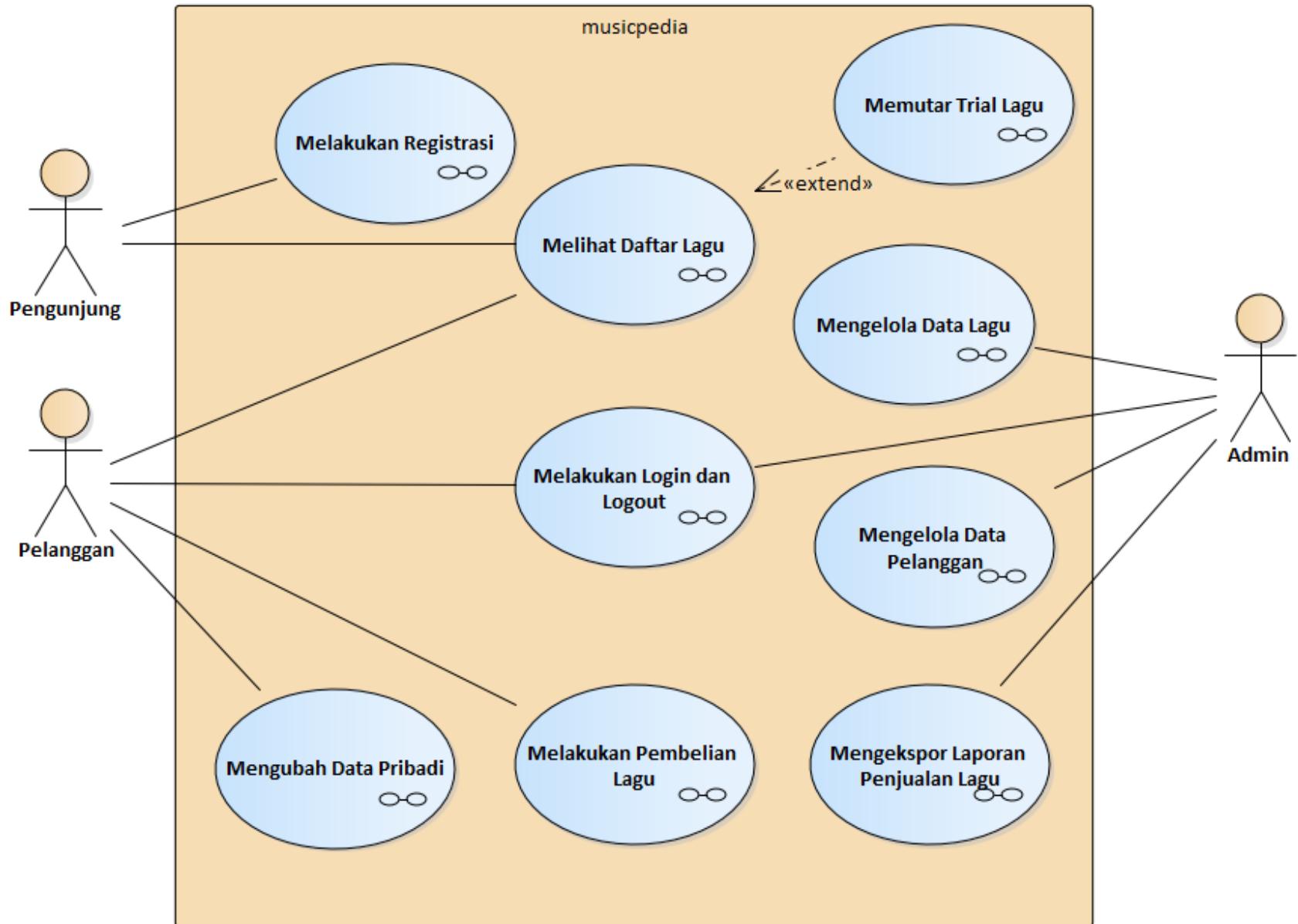


# Analysis

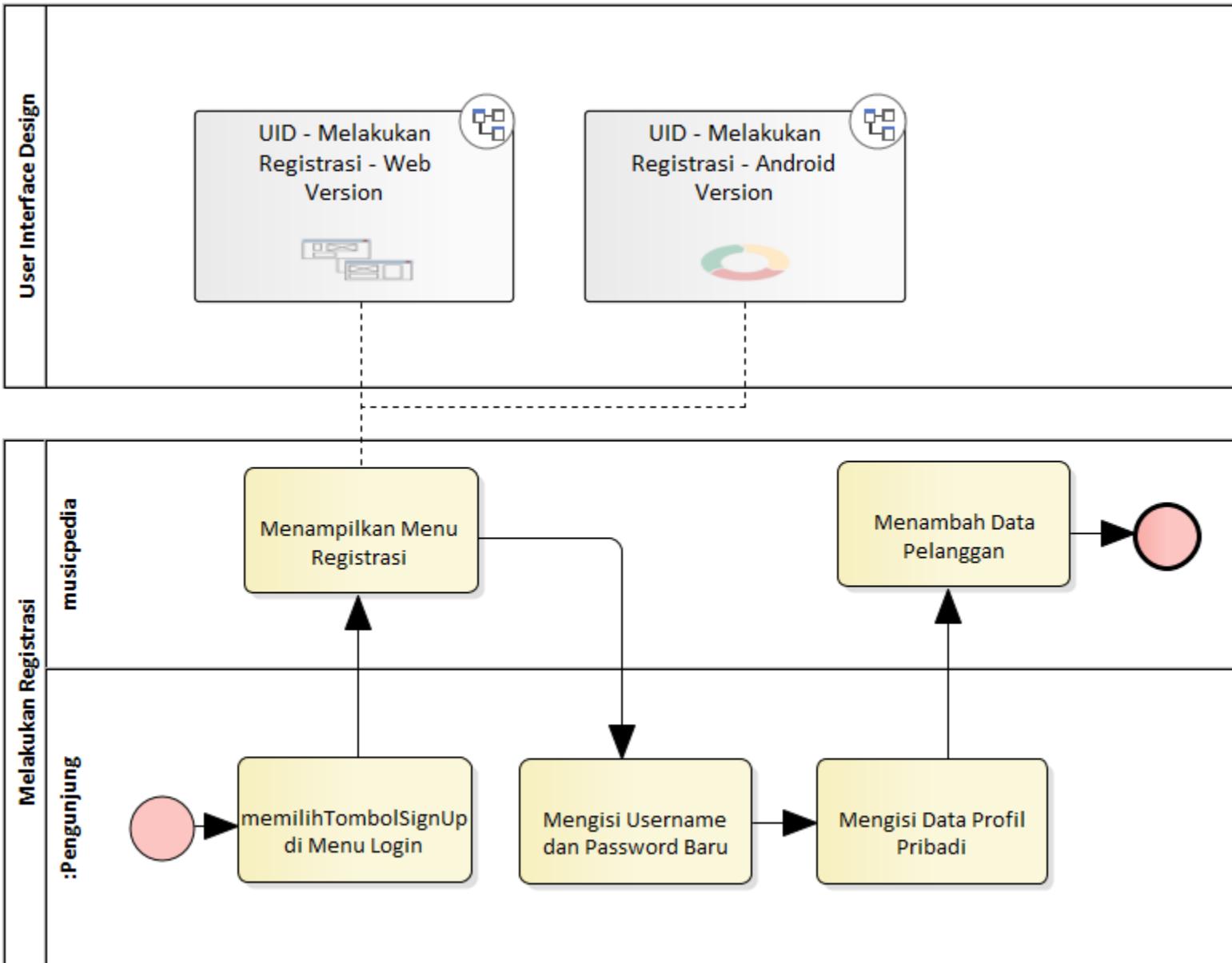
1. Pengumpulan dan analisis kebutuhan  
**(Requirements):**
  - Siapa yang menggunakan software?
  - Apa yang dilakukan oleh software?
  - Kapan software digunakan?
2. Investigasi Software yang Ada (**Baseline**)
3. Identifikasi **Peluang untuk Perbaikan**

**(System Specification)**

# Use Case Diagram MusicPedia



# BPMN Melakukan Registrasi



# User Interface Design Melakukan Registrasi (versi Web dan versi Android)

The image shows a side-by-side comparison of user interface designs for a registration form, one for a web browser and one for an Android mobile application.

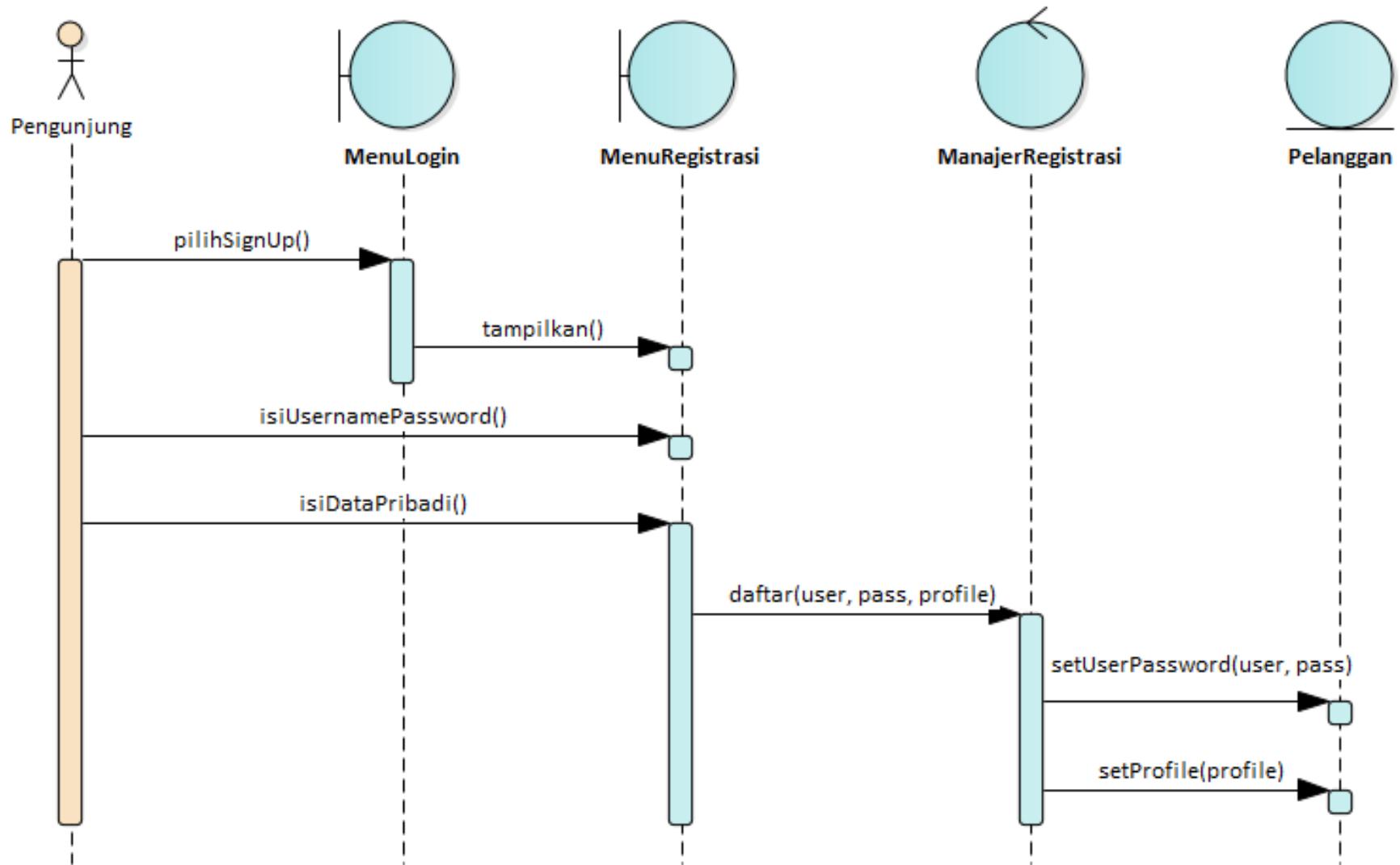
**Web Version (Left):**

- Header: Musicipedia Indonesia, Braindevs, www.musicpedia.com
- Form Fields:
  - Username
  - Password
  - Konfirmasi Password
  - Nama Lengkap
- Date of Birth Selection:
  - Hari (Day) button
  - Bulan (Month) dropdown menu showing Januari and ...
  - Tahun (Year) button
- Gender Selection:
  - Pria (Male) radio button
  - Wanita (Female) radio button
- Agreement Checkbox:  menyetujui Syarat dan Ketentuan Penggunaan
- DAFTAR (Register) button

**Mobile Version (Right):**

- Header: 11:18 PM
- Form Fields:
  - Username
  - Password
  - Konfirmasi Password
  - Nama Lengkap
- Date of Birth Selection:
  - Hari (Day) button
  - Bulan (Month) dropdown menu showing Januari
  - Tahun (Year) button
- Gender Selection:
  - Pria (Male) radio button
  - Wanita (Female) radio button
- DAFTAR (Register) button

# Sequence Diagram Melakukan Registrasi





# Design

## 1. Program Design (UML Diagrams)

- Software seperti apa yang ingin dibuat
- Komposisi dan arsitektur dari software

## 2. User Interface Design

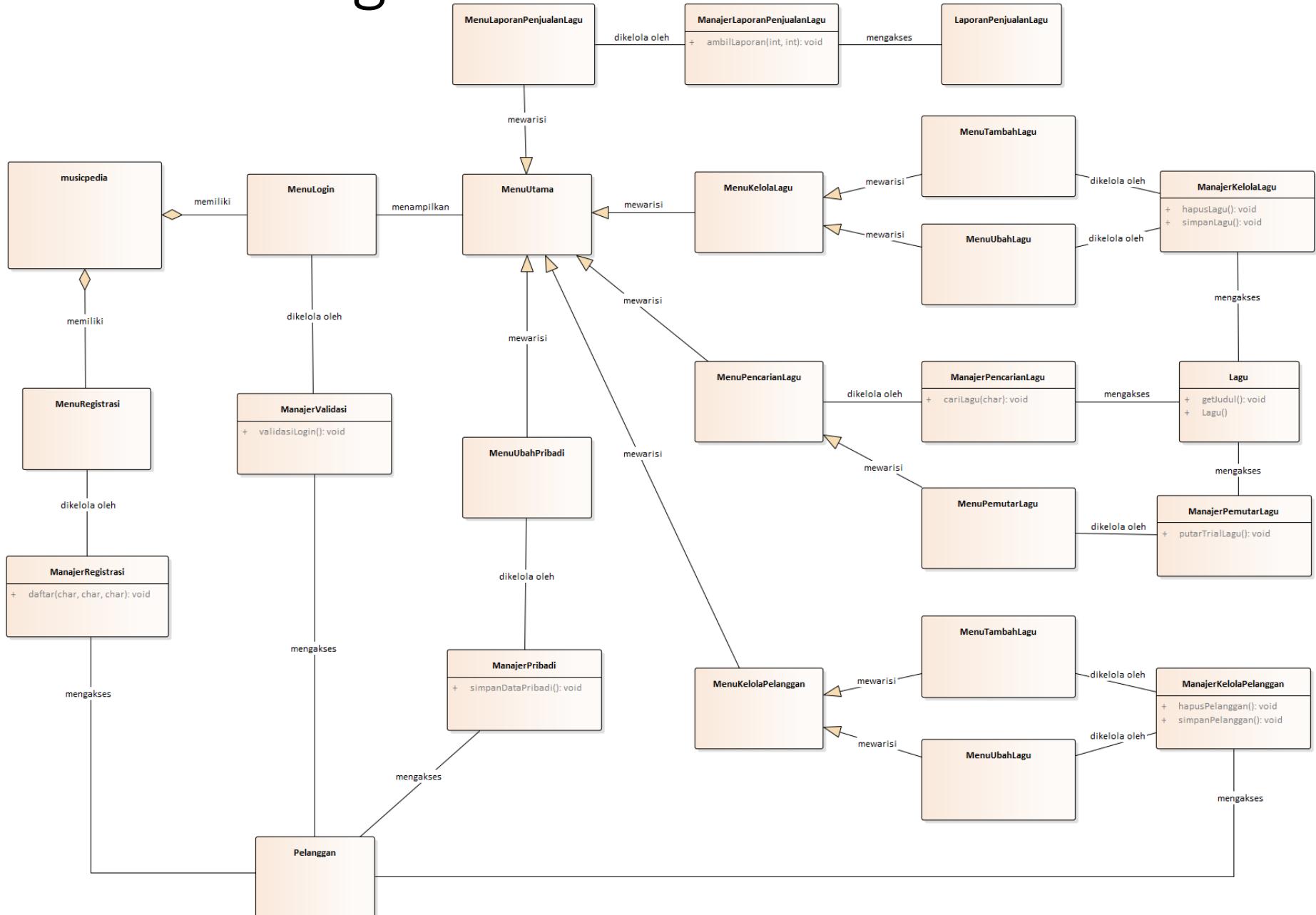
- Bagaimana pengguna berinteraksi dengan software
- Pahami form/laporan yang digunakan oleh perusahaan

## 3. Data Design

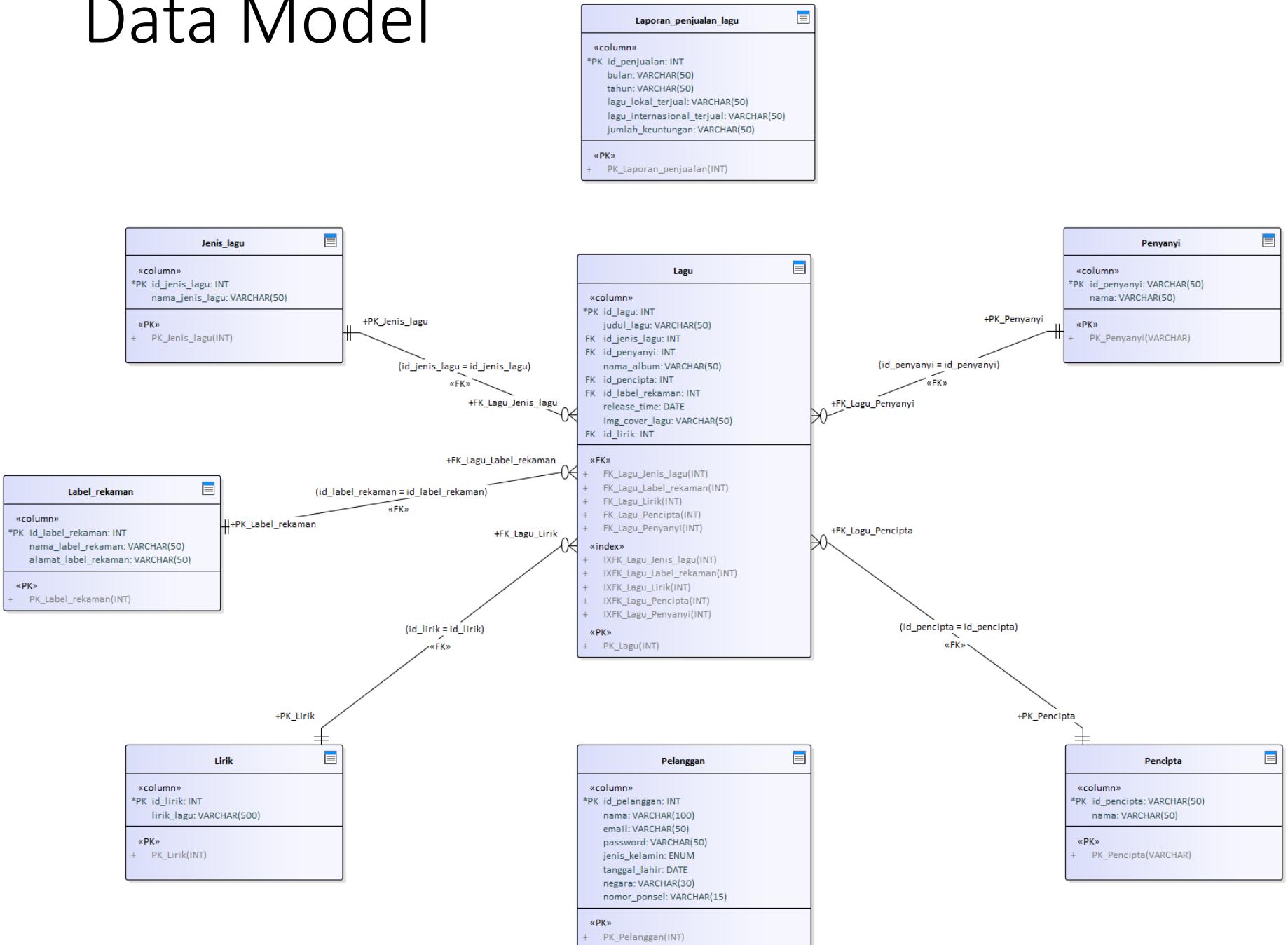
- Data apa yang akan disimpan
- Format data yang disimpan
- Dimana data akan disimpan

(**System Specification**)

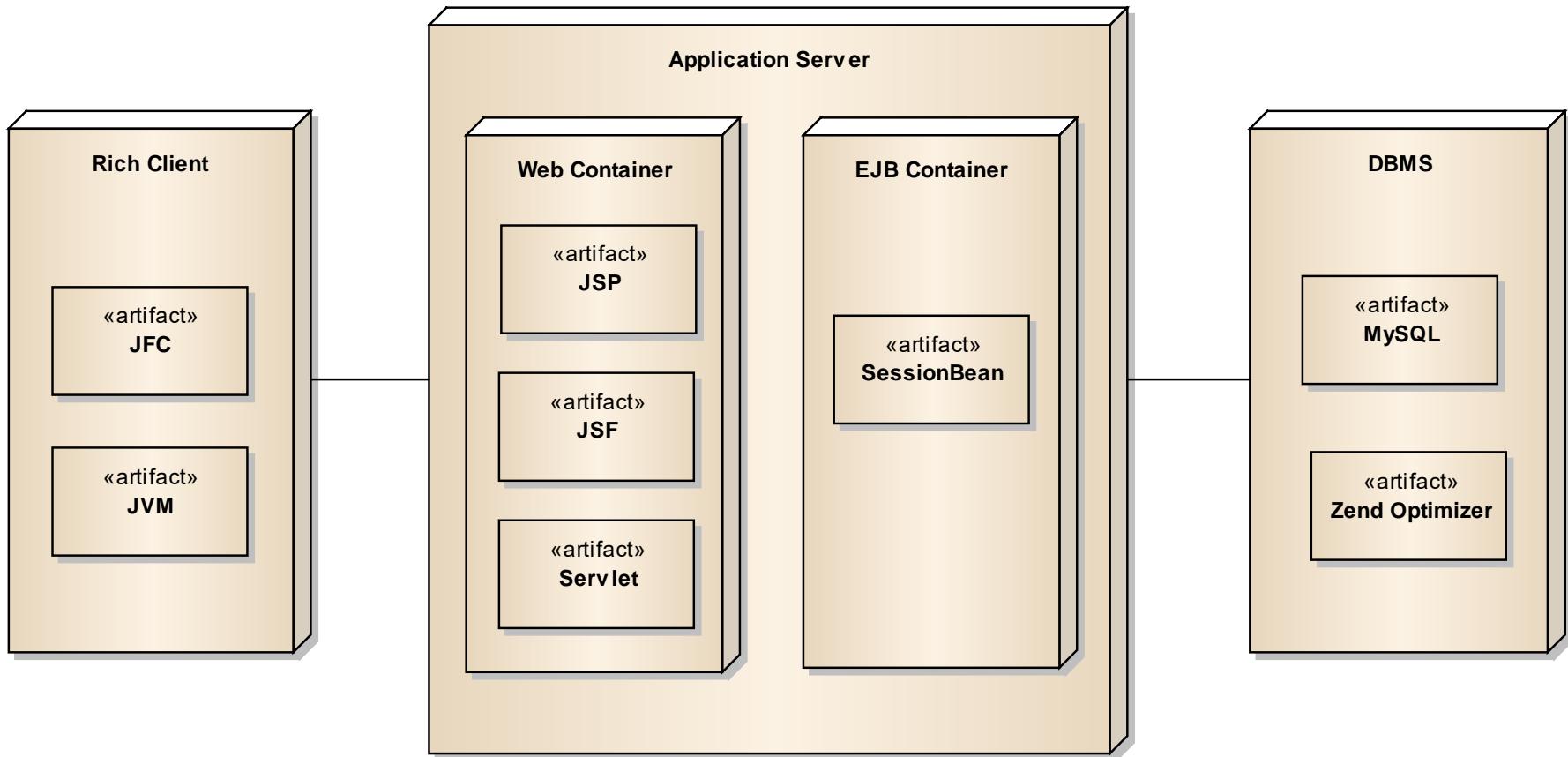
# Class Diagram



# Data Model



# Deployment Diagram





# Implementation

## 1. Konstruksi Software

- Pembuatan kode program

## 2. Pengujian Software

1. Unit Testing
2. Integration Testing
3. System Testing
4. User Acceptance Test

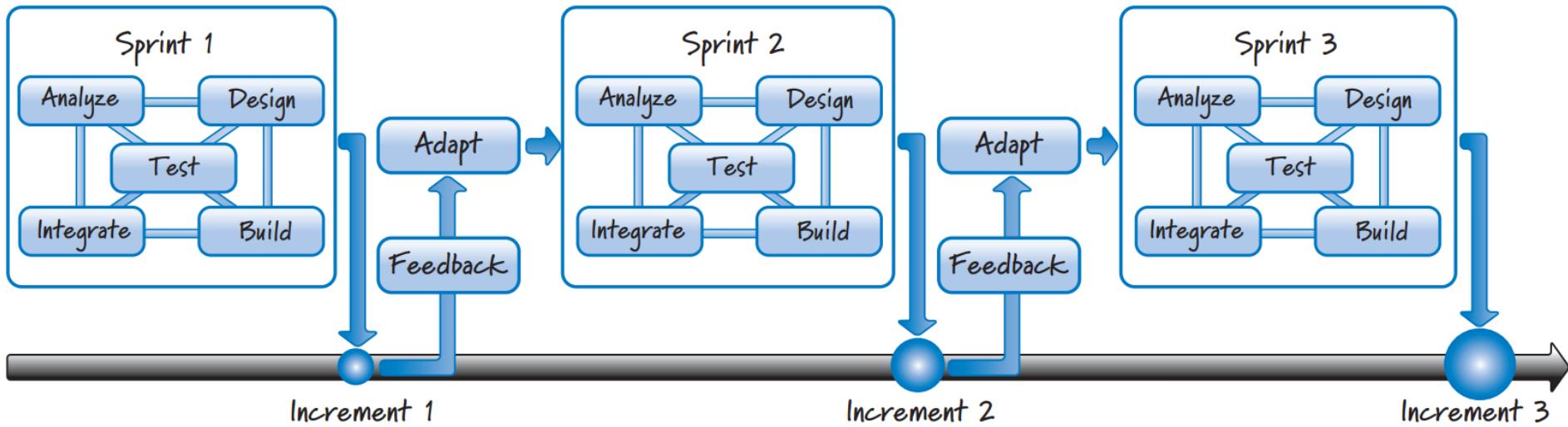
## 3. Dokumentasi

1. User Documentation
2. System Documentation

## 4. Installation

- Software lama dimatikan
- Software baru diaktifkan (instalasi)

# Sprint and Shippable Product Increment



## MITOS 1

Penerapan Scrum yang Masih  
**Bercampur dengan SDLC** Membuat  
Developers masih Mengikuti  
**Metodologi Waterfall**

# 3 MITOS

PENYEBAB KEGAGALAN

**PENERAPAN SCRUM**

## MITOS 3

Masih Memikirkan  
**Dokumentasi Pengembangan**,  
dan Kurang Fokus ke  
Pengembangan Produk

## MITOS 2

Tidak Mengikuti Panduan yang  
sudah Dibuat Scrum Guide dan  
Malah Membuat Improvement  
yang Akhirnya Gagal

# TERIMA KASIH

*Wahyu Utomo*

wahyu@brainmatics.com  
<http://brainmatics.com>  
08118228887





# References

1. Alan Dennis et al, **Systems Analysis and Design with UML 5<sup>th</sup> Edition**, *John Wiley and Sons*, 2016
2. Joseph S. Valacich and Joey F. George, **Modern Systems Analysis and Design 8<sup>th</sup> Edition**, *Pearson Education*, 2017
3. Scott Tilley and Harry J. Rosenblatt, **Systems Analysis and Design 11<sup>th</sup> Edition**, *Cengage Learning*, 2017
4. Kenneth E. Kendall and Julie E Kendall, **Systems Analysis and Design 8<sup>th</sup> Edition**, *Prentice Hall*, 2010
5. John W. Satzinger, Robert B. Jackson, Stephen D. Burd, **Systems Analysis and Design in a Changing World 6<sup>th</sup> Edition**, *Course Technology*, 2012
6. Hassan Gomaa, **Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures**, *Cambridge University Press*, 2011
7. Howard Podeswa, **UML for the IT Business Analyst 2<sup>nd</sup> Edition**, *Course Technology*, 2009
8. Jeffrey A. Hoffer et al, **Modern Systems Analysis and Design 6<sup>th</sup> Edition**, *Prentice Hall*, 2010
9. Albert Endres and Dieter Rombach, **A Handbook of Software and Systems Engineering**, *Pearson Education*, 2003
10. Ken Schwaber, Jeff Sutherland, **The Scrum Guide**, <http://scrumguides.org>, 2020
11. Kenneth S. Rubin, **Essential Scrum A Practical Guide To The Most Popular Agile Process**, Addison Wedley, 2013
12. Henrik Kniberg, **Scrum and XP From The Trenches 2<sup>nd</sup> Edition**, InfoQ, 2015
13. Andrew Pham and Phuong-Van Pham, **Scrum in Action Agile Software Project Management and Development**, Cengage Learning, 2012
14. Joakim Verona, **Practical DevOps**, Packt Publishing, 2016
15. Mitesh Soni, **DevOps for Web Development**, Packt Publishing, 2016