



Code Development

Ridi Ferdiana
ridi@acm.org
Version 1.0.0

Before do the Code

- The Requirements
 - What we need to build
 - What constraint that we have
- The Analysis
 - How the business works
 - How the solution will help to solve the problems
- The Architecture
 - How the solution is deployed
 - How the solution is composed
 - How the solution is 'look and feel'

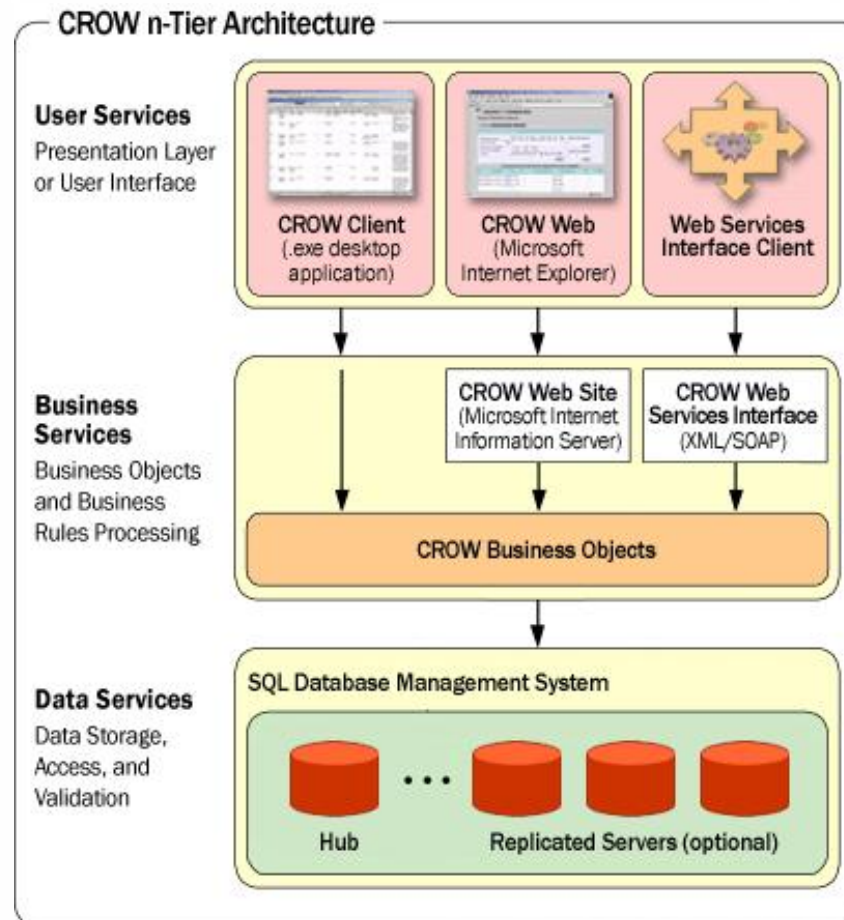
Codes Development Required Skills

- Software Architecture
- Development Infrastructure
- Estimation Technique
- Developer Guidences

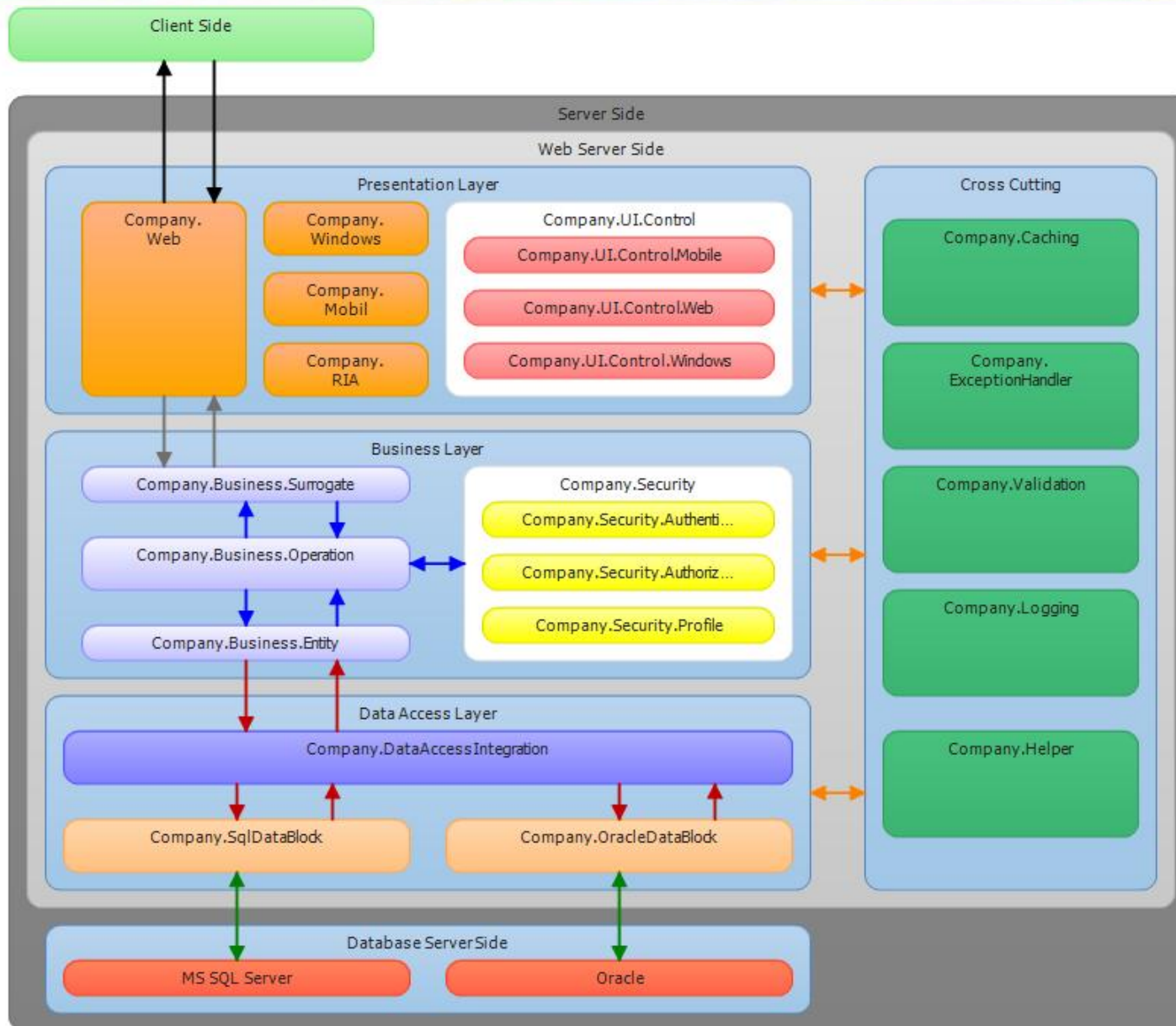
Typical Architecture

- Program Organization
 - i.e. Tier Architecture
- Major Classess
 - i.e. Layered Architecture
- Data Design
 - i.e. Data Driven Architecture
- Business Rules
 - i.e. Service Oriented Architecture
- UI Design
 - i.e. MVC, MVVM, MVP

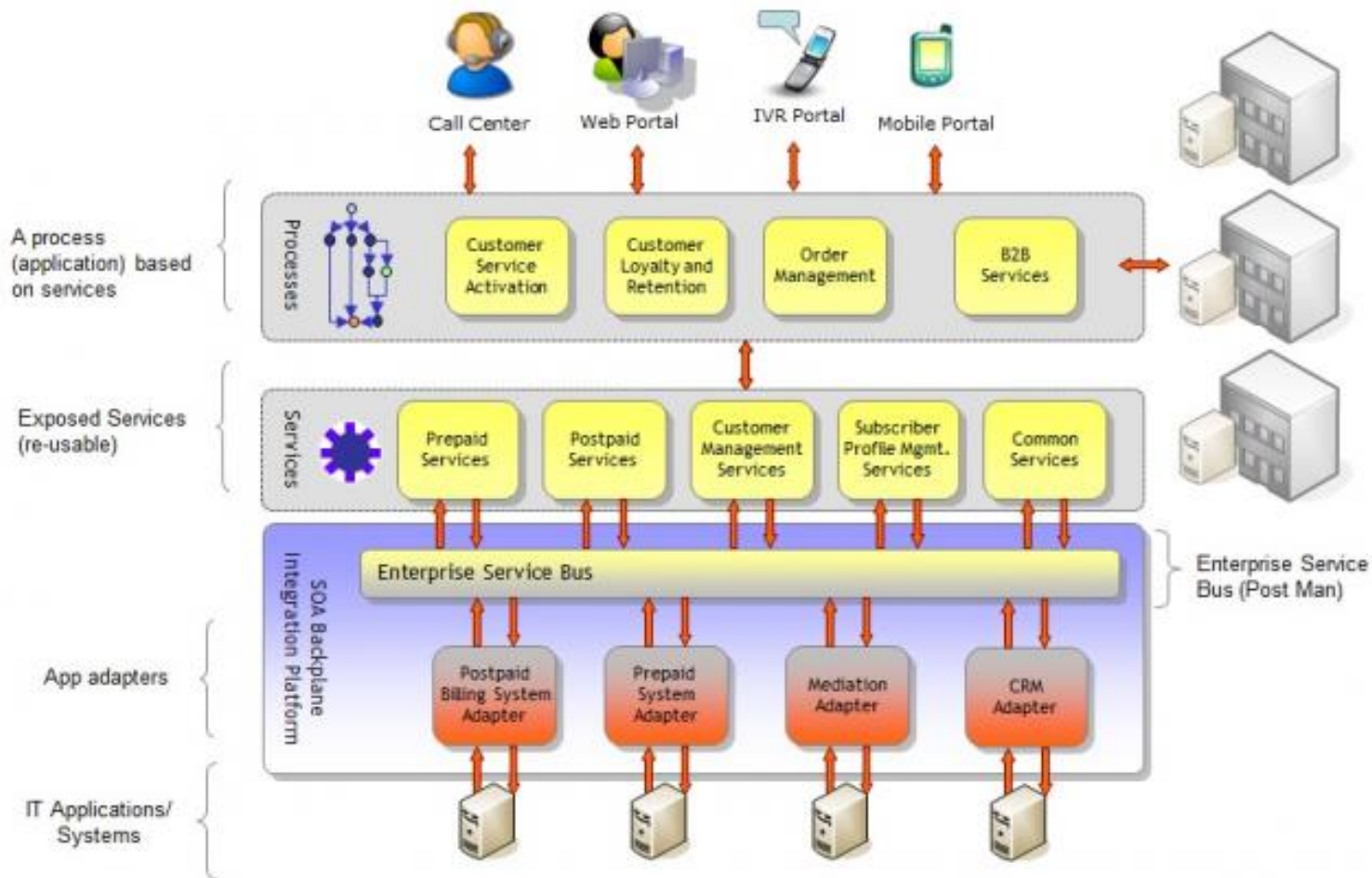
Tier Architecture



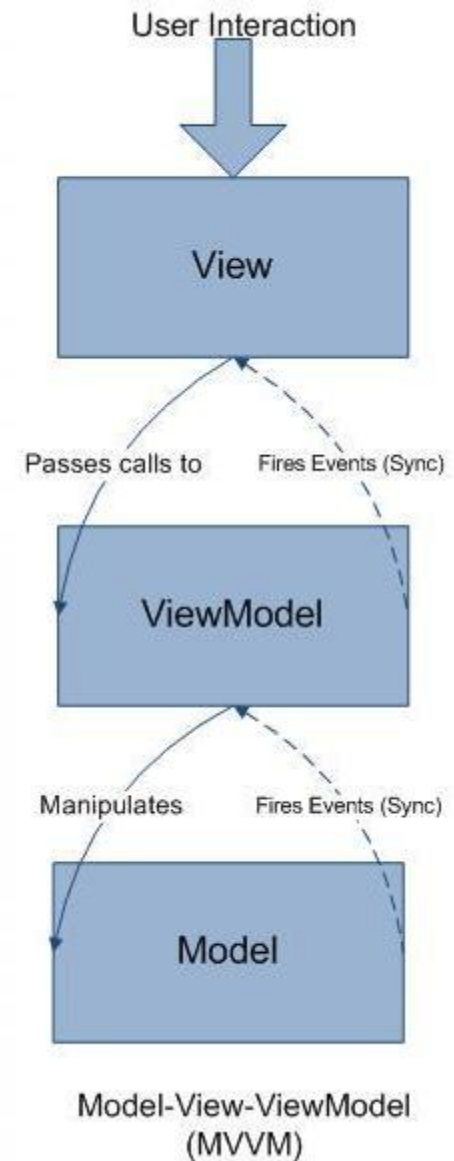
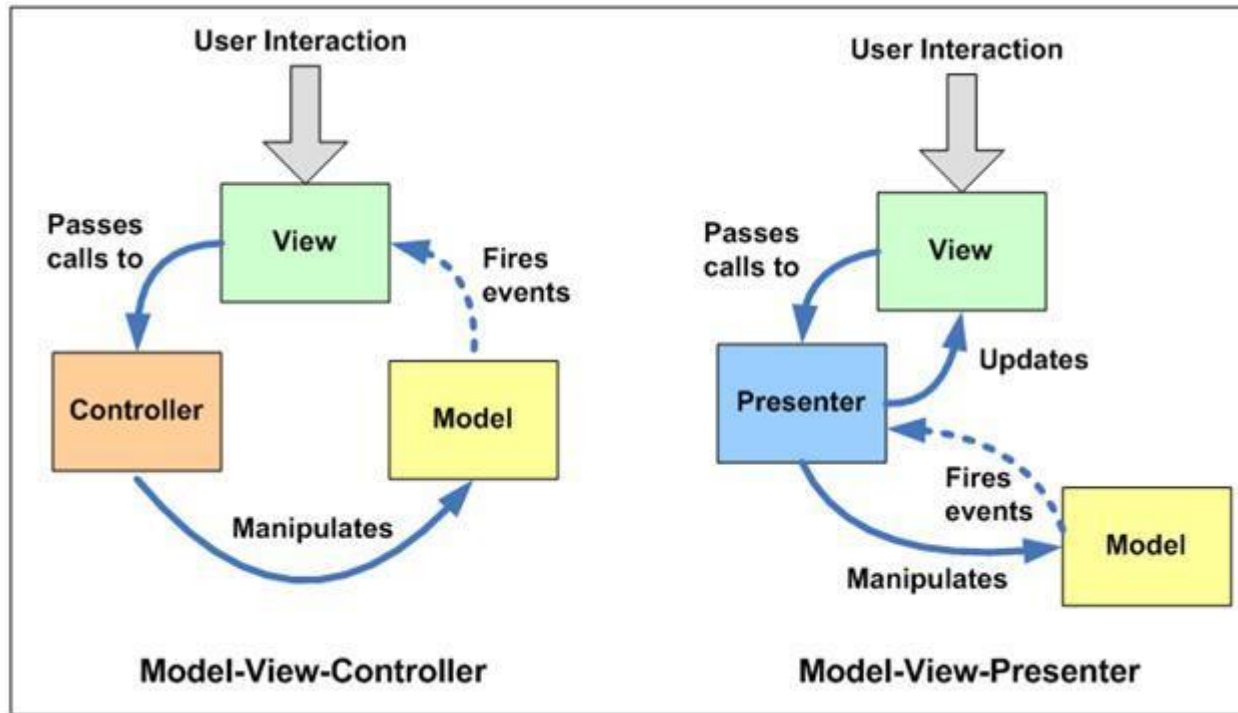
Layered Architecture



Service Oriented Architecture



MVC, MVVM, MVP





DEMO

Data Driven Architecture

Building Line of Business Application
with Data Driven Arch / Programming

Development Infrastructures

- Source Code Versioning (SVN, TFS, GIT)
- Project Management Portal (TFS, VersionOne, Rally, TeamPulse)
- Integrated Development Environment (Visual Studio, Net Beans, Eclipse)
- ALM Solutions (MS-ALM, CLM)
- Collaboration and Communication Portal (Office 365, Google Apps, Cisco WebEx)
- Development Server

Ideal way to estimate

Estimating by the team

Estimation should be done by the team who will do the job

Using Historical Analysis

Past performance for future prediction

Estimation Techniques

Specific and Scientific Technique, i.e. LOC, COCOMO, Function Points, etc



Pragmatic way to estimate

- Count what you can count

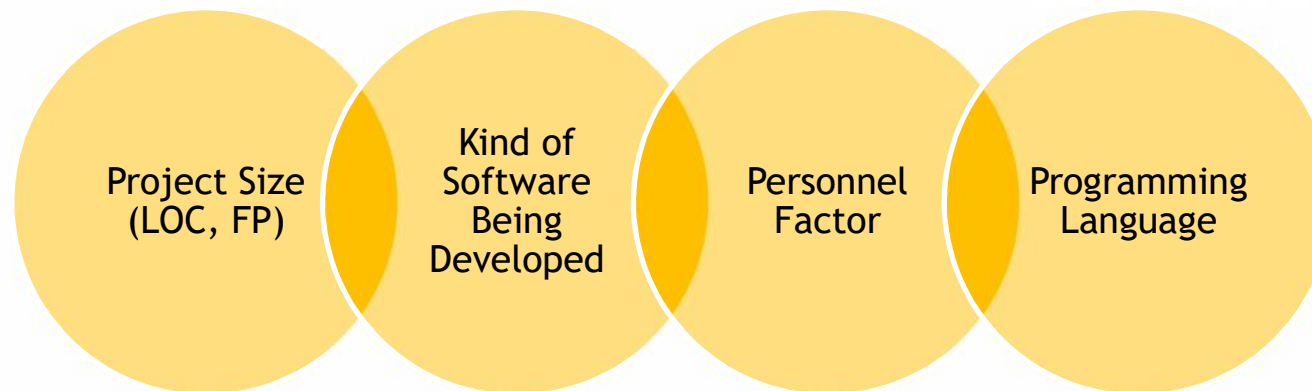
Item	Unit
Features	Effort hours / feature
Stories	Effort hours / story
Function point	LOC / function
Change request	Effort hours / changes
Web Page	Effort hours / web page

Estimating Undercover

- Wati: “Hey Budi how long to develop module x?”
- Budi to himself: “argh, I’m busy. That will take two days but I can’t afford to lose”
- Budi : “about a week”
- Wati to herself: “he always say that so 2.5 multiplication rules”
- Budi to himself : “I can stall those out for week“



Estimate Influence (McConnell, 2006)



Estimation after Requirements Engineering

- **Formal Process**
 - Using Software Metrics (Use Case Points)
 - Project Management Approaches (Resources, Budget and Time)
- **Agile Process**
 - Using user story estimation



DEMO

User Story Estimation

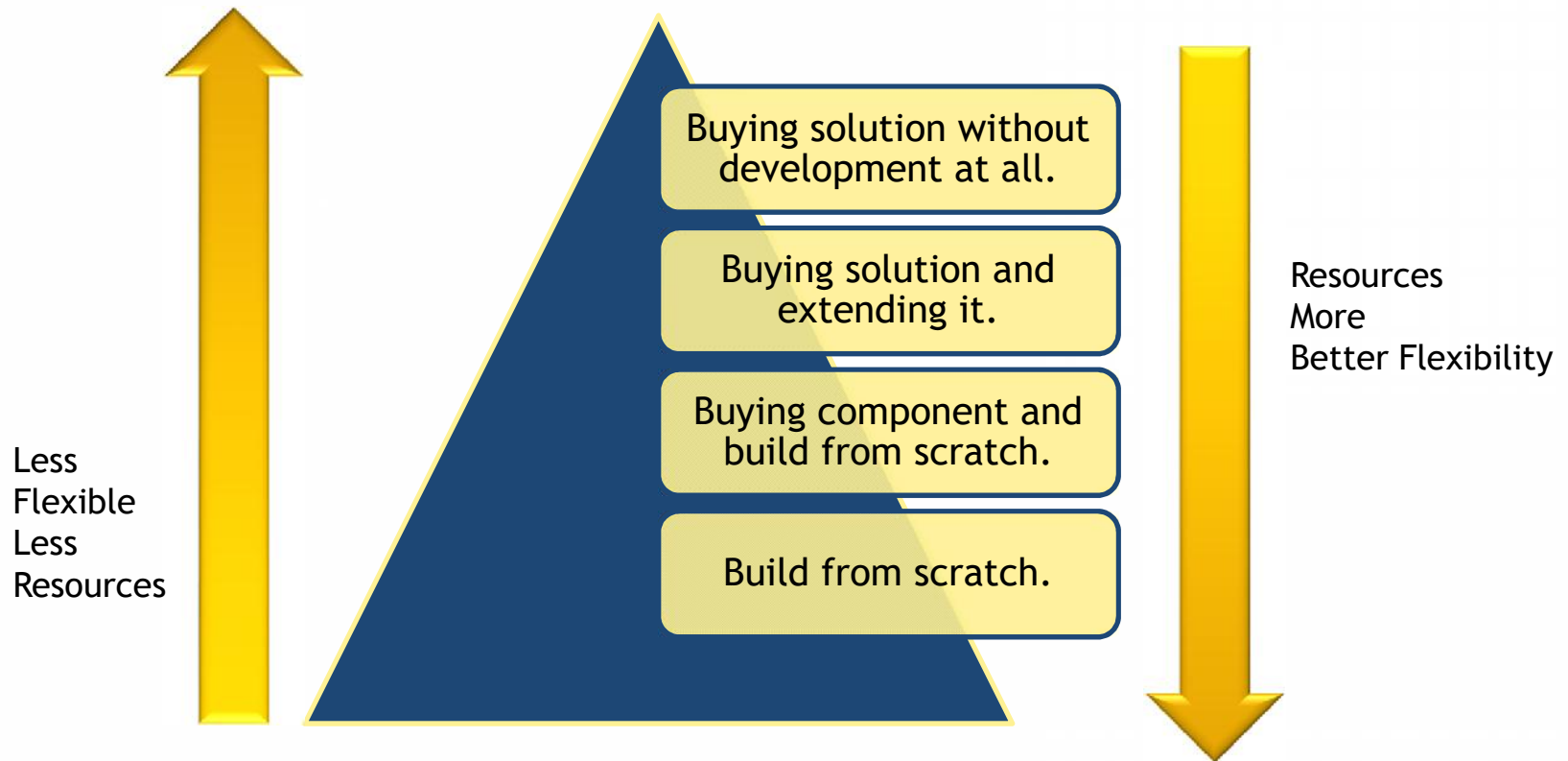
Planning Game on Agile Software
Development

Development Guidences

- Buy or Build Solution
- Choices of Programming Technology
- Coding Standard
- Development Practices
- Recommended Team Structure
- Technology Wave Adoption
- Versioning Strategy

Developer
LogBook

Buy or Build



Choices of Programming

One language for ALL: i.e. Java, Javascript

One Library for Many Language: i.e. .NET Framework

One Language for One Platform: i.e. PHP, ActionScript

Coding Standard

- Code Comment : XML, Inline comment
- Coding Conventions: Camel case, Pascal case, Hungarian case
- Developer Notes: developer flag i.e. TODO, NODO, HACK
- Build Notes
 - Readme
 - Refactoring
 - Batchscript

Development Practices

- Development process
 - Featured-Driven Development
 - Test-Driven Development
 - Behavior-Driven Development
- Development management
 - Pair Programming
 - Parallel mode



Recommended Team Structure

- Team structure is depend on the method that you are using (e.g. Extreme Programming, SCRUM, UP)

	Product Management	Program Management	Architecture	Development	Test	User Experience
Program Management	N					
Architecture	N	P				
Development	N	N	P			
Test	P	U	U	N		
User Experience	P	U	U	N	P	
Release/Operations	U	U	U	N	U	U

P Possible **U** Unlikely **N** Not Recommended

Technology Wave Adoption

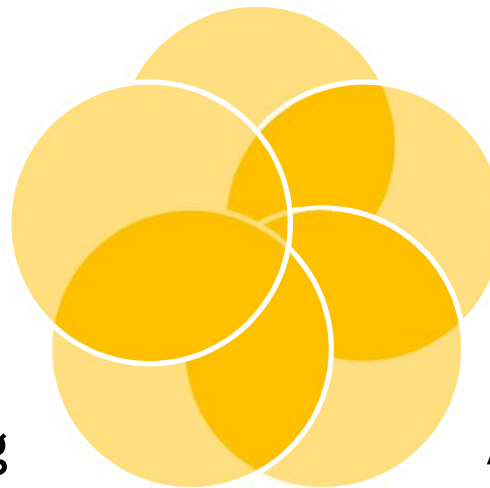
Types of
Application

Data
Storage

The use of
Application
Server

Messaging
Concept

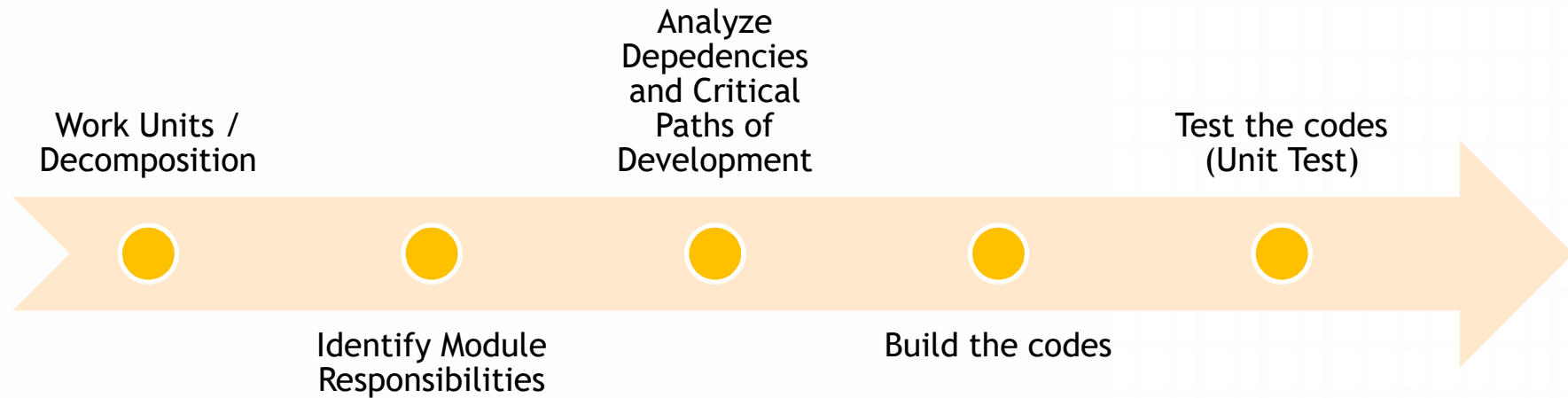
Application
Library



Versioning Strategy

- Product Jargon versioning
 - CTP → Alpha → Beta → RC0 → RTM/RTW
- Numeric versioning (v.1.2.5)
 - Major version (1.x.x) → project
 - Minor Version (x.2.x) → phase oriented
 - Revision version (x.x.5) → successfully build
- Build versioning
 - Build date (e.g. Windows7-020310)
 - System Build (e.g. Windows7-Build7600)

5 Practical steps to build codes



Define Work Units

the decomposition of your system

PIC team composition

an identification of the modules that make up the system,

sort of size estimate

description of the responsibility for each module

Identify Module Responsibilities

- Define the functions of each modules
- Abstract Data Types that need to be agreed
- Can be displayed as a set of sequence diagrams describing how the requirements are realized by the system

Analyze Dependencies

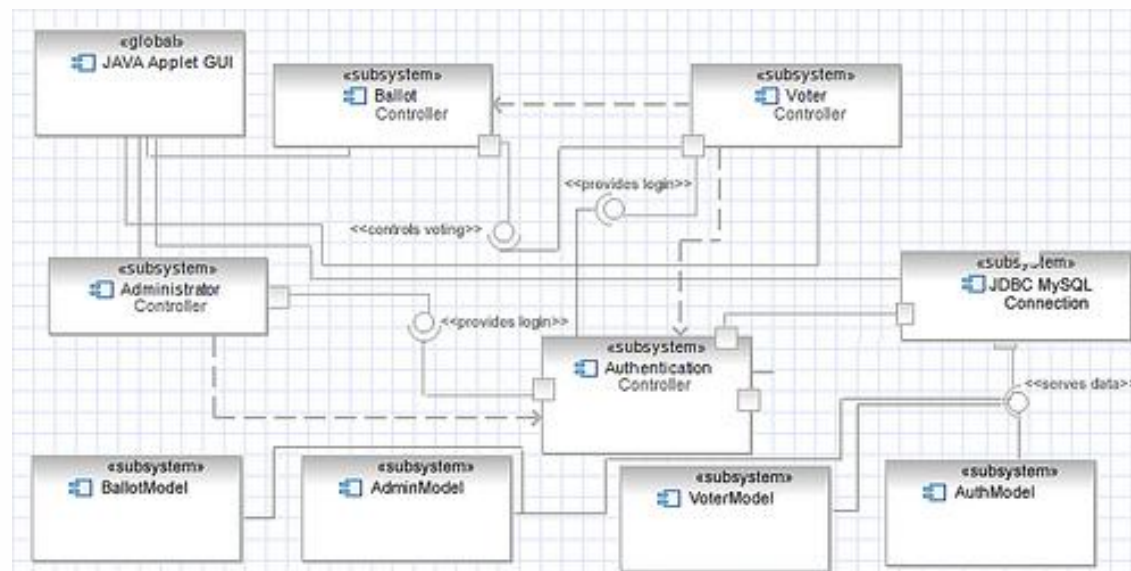
		1	3	14	4	5	6	15	9	11	16	12	7	10	13	8	2	Number of modules this depends on
Publish Subscribe	1																	0
Command Processing	3																	0
Configuration	14																	0
Value Cache	4	1																1
Data Access	5				1													1
Condition Evaluation	6					1												1
COV Processing	15		1															1
Adapter Manager	9	1	1	1		1		1										5
Property Display	11	1				1			1									3
Alarm Processing	16	1				1			1									3
Alarm Rule Engine	12	1				1	1				1							4
Hierarchy Editor & Display	7					1				1								2
Alarm Display	10					1					1							2
L&R Rule Engine	13	1				1	1		1									4
Rule Editor	8					1						1				1		3
Logon	2					1				1			1	1				4
Number of dependants		6	2	1	1	10	2	1	3	2	2	1	1	1	1	0	0	

Legend

- 1 in bold = uses dependencies
- 1 regular = regular dependencies

Build codes

- Data Access Development
- Business Logic Development
- Service Development
- User Interface Development



Test the codes

- Run through the codes through high fidelity prototyping
- Debugging activity
- Creating unit test

History of Software Testing



Key points

- Code development is the key of software development activity
- Programmer should considers to build developer log book that covers
 - Common Patterns of Software Architecture
 - Common way to estimate the software
 - Development Guidences
- 5 practical ways to design and to build codes

References

- Steve McConnell. 2004. Code Complete. Microsoft Press
- Niraj Bhatt. 2009. MVC vs. MVP vs. MVVM. Architect's blog.
<https://nirajrules.wordpress.com>
- Ridi Ferdiana. 2014. Application Lifecycle Management Course Module. MCT Module