-

,

, ,
Product

AbstactFactory, ConcreteFactory, AbstractProduct, ConcreteProduct,Client

client , Product , Product

.

-

ADT ,
API

Builder= API, ConcreteBuilder= , Director= workflow, Product= ADT

Product , Product

vs

,

, ,

-

UI event handling

msg receiver , msg

Handler(Handler), ConcreteHandler

msg receiver, sender (= /
)

msg ,

-

Container

container , API
,

Aggregate(Container ADT), ConcreteAggregator, Iterator, ConcreteIterator

- Command

, ,

Command(API), ConcreteCommand, Invoker( ), Receiver(client)

, , ,undo

-

client ,

Product, ConcreteProduct, Creator, ConcreteCreator

.

Product Creator .

- Adapter

client ,

ObjectAdapter( ), ClassAdpater( ) 2 . Target= , Adapter, Adaptee( )

Adapter vs Proxy

Adapter interface vs proxy realSubject redirection

Proxy .

-

.

, ,

Abstraction, RefinedAbstraction(client API), Implementor(ADT), ConcreteImplementor

, ,

Adapter vs Bridge

. . client . Bridge
.

- Mediator

workflow .

interaction ,
interaction

Mediator, ConcreteMediator, Colleague, ConcretreColleague

Mediator , ,
, Mediator

Facade vs Mediator

Facade Interface vs Mediator interaction .

Facade vs Mediator .

- State

, if-else

Context(client API), State, ConcreteState

state , State state
.

- Composite

, API

Component(client API), Leaf( ), Composite(Component API)

client , hierarchy

(1 ,)

- Visitor

.

, ,
,

Visitor, ConcreteVisitor, Element, ConcreteElement

,

ConcreteElement Visitor .

- Observer

Subject(Observer , notify() observer), ConcreteSubject, Observer, ConcreteObserver(subject ConcreteSubject notify .)

1:N sender, receiver , ,

- Strategy

, ,

Context(client API), Strategy, ConcreteStategy

, ,

, ConcreteStategy Stategy , .

SOLID - Single Responsibility Principle

.

? . .

SOLID - Open Closed Principle

, .

SOLID - Liskov substituion principle

SOLID - Interface Segregation principle

? .

? (mixin, interface), Delegation

SOLID - Dependency Inversion Principle

' , . (ADT) ( ) ? .

Coupling vs Cohension

-> (, Composition,)

-> . , .

	System Service (FR) Context(NFR)
SDLC	Water fall, Incremental, Evolutionary, Spiral Iterative, Agile, RUP
RUP	Inception, elaboration, construction, transition
Incremental	RE RE , code code , Test Test
Validation vs Verification	Verification , Validation

(cont)

RE	Feasibility Test > RE Elicitation > Req Specification > Req Validation
Elicitation	System model Workshop, Brainstorming, survey, interview, role playing, prototyping . Prioritization&Negotiation NFR . (FR)
Negotiation	ranking, ahp, grouping, bubble sort, hundred dollors
Specification	/ .
SRS	Software Requirement Specification ( ). Plan .
Validation	( )

- QA - Functionality

Suitability( )

Accuracy

Interoperability( ) interaction

- QA - Functionality (cont)

Security , ,

Compliance( ) SW , ,

- QA - Reliability

Maturity( ) SW

Fault Tolerance(,)

Recoverability( ) ,

- QA - Usability

Usability

Understandability( ) , ?

Learnability( )

Operability( ) SW / ?

Attractiveness( )

- QA - Effectivness

Effectiviness ?

TimeBehavior( ) ?

Resource Utilization( )

- QA - Maintainability

Analyzability( ) SW

Changeability( )

Stability( ) SW

Testability SW

- QA - Portability

Adaptability(SW)

Installability( )

Co-existency(, SW)

- QA - Portability (cont)

Replaceability( ) SW

- Viewpoints 4+1

Viewpoints?	. View . Logical Views , Process View , , Development (,) , Deployment
Use-Case view	Activity/Info Flow Diagram, UseCase diagram
Logical View	Sequence Diagram, Domain Model Class Diagram
Development view	component diagram, package diagram
Domain Model?	RE Conceptual class diagram.
Process view	sequence, activity diagram
Physical view	deployment diagram

Quality Life cycle

Process Quality	
Internal QA	QA
External QA	Stake holder QA. 2
Quality in use	QA ( )

Quality Life cycle (cont)

Architecture concerns	, .

- GRASP

Responsibility ?	contract . / . , private field
GRASP	coupling, cohension, expert, controller, creator, don't talk to stranger
Low coupling ?	. , ,
cohension ?	,
Information Expert	, private info (encapsulated) .

- GRASP (cont)

Creator	A B , B , B aggregate B A B .
controller	
Don't talk to stranger	

- ADD

ADD?	Architecture Design Methodology
Plan do and check	-> -> Architecture Driver -> NFR Style + FR -> -> Usecase, QA contraint ->

- Unified process

UP	inception phase, elaboration, construction, transition
phase disciplines	test, code, planning, Req, analysis & design, business modeling, deployment, configuration & change management, Project management
Inception phase	Business modeling, Req , Prototyping, feasibility study

- Unified process (cont)

Elaboration	ADD Baseline Architecture . Baseline archirecture skeleton architecture elaboration .

- Pattern, Style, Idium

Architecture Style	== Architecture Pattern
Pattern?	(,) .
Design pattern?	components subsystem
idium	

Association	
Aggregation	owner owning .
Composition	.

Source( ), (Stimulus), (Artifact), (Environment), (Response), (Response Measure)

(cont)

Modifiability	( ), , LateBinding
( )	(, , ,
( )	, , (HW),
( )	Ping,Echo(,), HeartBeat( ),
( )	/Node ,
	( ), (write), ( ), Non-Repudication(, , AccessControl( ), ( )

UML

class diagram	class, interface, component, visibility
deployment diagram	artifacts(file, db), node, components
component diagram	port, provided interface, required interface, component,
sequence	lifeline, message, interaction use,
activity diagram	swim lanes, merge node, decision node, initial node, final node, merge node, fork node
data flow	external entity, process, datastore

- Batch Sequencial

Data Flow

Data Flow AS .
. input .

Program, Data Store

( ) ,

input .
latency .

- Pipe and Filter

Data Flow

- Pipe and Filter (cont)

.

, ,

.

., , , .

.

Batch , &
Sequencial .

- Process Control

Data Flow

Controller, Actuator, Process, Sensor

Controller Actuator start/stop . Actuator Process Process input start/stop . sensor Controller . closed - loop .

- Process Control (cont)

.

, ,

- Repository

Repository, Components

. interaction .

, BackUp&Restore , components / (interaction),

, components , .

- Blackboard

Data Centered

Blackboard, Control, KnowledgeSource

KnowledgeSource 1 . Blackboard KnowledgeSource blackboard . Control .

- Blackboard (cont)

?

KnowledgeSource . .

blackboard . . KnowledgeSource

- Layered

Hierarchical

Layer

. . Layer layer .

?

, , layer / .,

- Layered (cont)

. , Exception Handling depth .

- PlugIn Architecture

Hierarchical

CoreSystem, Plug-in modules

App PlugIn . coreSystem .

, ,

- Virtual Machine

Application, Virtual Machine, Existing Platform

, SW

- Non-buffered event invoc

Implict asynchronous communication software

Event source, event listener

interaction . .

listener / , (source, listener),

listener , ,

- Buffered Message Based

Implicit Asynchronous communication

MessageBroker, Consumer, Producer

Producer msg xml broker . Broker , borker route consumer .

, ,

Producer Consumer idle , / , producer ,

, , .

- Client - server

Distributed

Client, Server

tier .

- Client - server (cont)

	, Control ?
	. (presentation, businiess logic), ,

- Multi tier

	Distributed
	Tier
Layer vs Tier	Layer , Tier
	tier , , , ,

- Dispatcher

	Distributed
	Client, Broker, Server
	Server . Broker Server load Server client . Broker tier , Server Broker register .
	Client-Server , , ,

- Dispatcher (cont)

	Broker , .

- Broker

	distributed
	Broker, Client, Servant
	CORBA . Node Broker Broker Node Servant . Node Node .
	, interaction ,
	, (fault-torelance),
	, , ? ....

- Master slave

	Distributed
	Master, Slave
	Master Slave
	fault tolerance, (slave)
	fault tolerance, reliability

- P2P

	Distributed
	Peer Provider Consumer contents
structured, unstructured	unstructured , structured .
	, , DB
	, node power

- Service oriented

	distributed
	(OS) MVC . XML . . XML .

- Service oriented (cont)

,
?

, ,

, , .

- micro service

	Distributed
	SOA . SOA monolithic . interaction . (DB) Microsoervice API encapsulation . DB DB .
RESTful?	API . json https . RESTful API HTTP GET, POST . API .
, . , / .	

- micro service (cont)

, ,
? micro service .

- MVC

View Control , Model impact .

- PAC