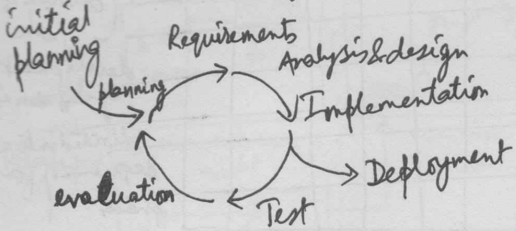
The Rational Unified Process – an introduction

:: S/W dev. Best practices :: — (commercially proven approaches to S/W development successfully)

1. develop S/W iteratively & incrementally; —



(each iteration results in an executable release)

2. Manage Requirements
3. use component-based architectures (with supported platforms; e.g. COM, CORBA, EJB)
4. visually model software
5. continuously verify software quality
6. control changes to software

} RUP (Built in)

* The Rational unified process model is built on three fundamental entities: (static structure)

- workers (roles), e.g., system analyst, designer, test ~~engineer~~ designer
- activities (unit of work performed) e.g., plan an iteration, find use cases and actors, review the design, execute a performance test
- artifacts (piece of information produced, modified, used by a process) e.g., a design model, a project plan, a defect, a project requirement database

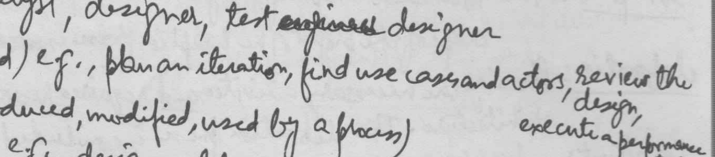
* workflows relate activities and workers in sequences that produce valuable results.

= core workflows (process)
- core engineering work flows (Business modeling, requirements, analysis & design, implementation, test, deployment)
- core supporting workflows (project management, configuration & change management, environment)

* ~~Guiddin~~ Guidelines, templates, and tool mentors complement the description of the process by providing detailed guidance to the practitioner.

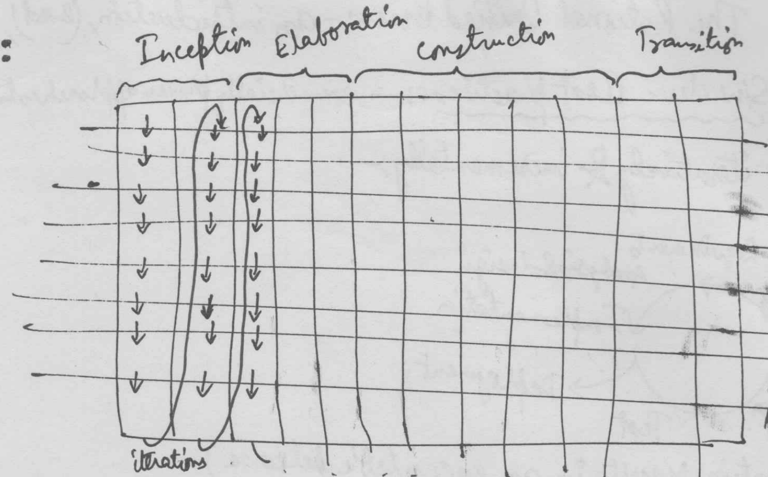
* Rational Unified Process is a process framework that is organized to enable the configuration of its static structure

* (Dynamic structure : Iterative development) :



* iterative processes are experimentations itselves ('iterative experimentations)

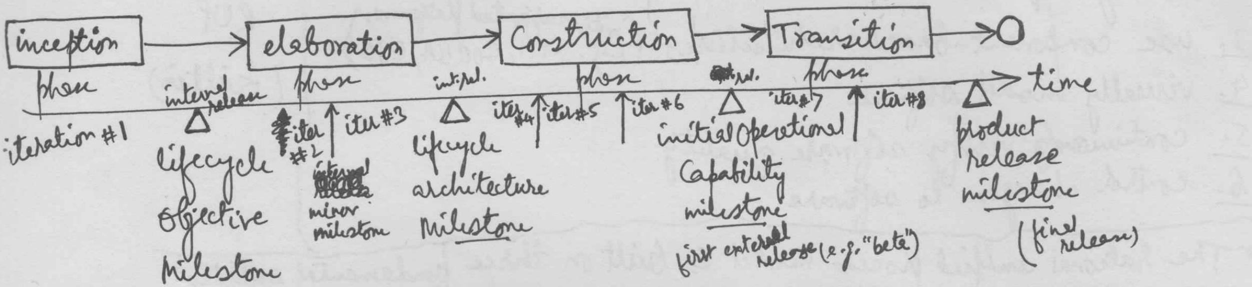
Dynamic structure: iterative development



✓ (divide & rule)

development cycle ~~is~~ is divided into a sequence of four phases

= Gaining Control: phases and milestones—

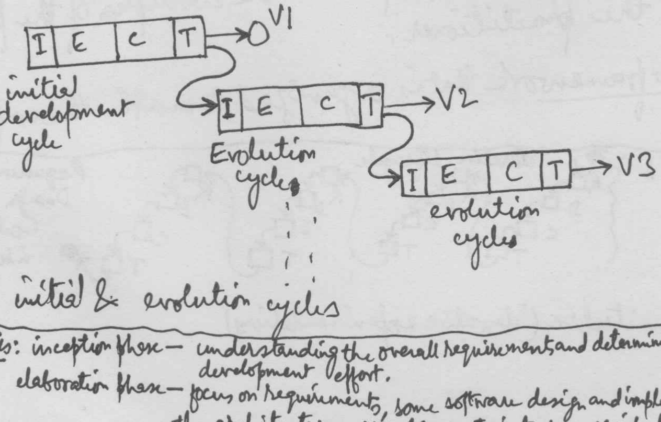

inception phase
elaboration phase
Construction phase
Transition phase
time
iteration #1
internal release
lifecycle Objective Milestone
iter #2
~~internal~~ minor milestone
iter #3
int. rel.
lifecycle architecture Milestone
iter #4
iter #5
iter #6
~~int.~~ rel.
iter #7
initial Operational Capability milestone
first external release (e.g. "beta")
iter #8
product release milestone
(final release)

Inception: The good idea— specifying the end-product vision and its business case and defining the scope of the project. The inception phase is ~~conc~~ concluded by the lifecycle objective (LCO) milestone.

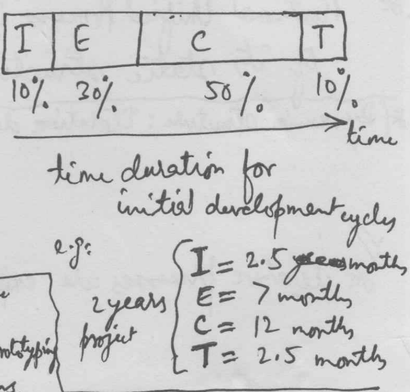
elaboration: Planning the necessary activities and required resources, specifying the features and designing the architecture. The elabration phase is concluded by the lifecycle architecture (LCA) milestone.

Construction: building the product and evolving the vision, the architecture, and the ~~pl~~ plans until the product—the completed vision—is ready for delivery to its user community. The construction phase is concluded by the initial operational capability (IOC) milestone.

Transition: Transitioning the product to its users, which includes manufacturing, delivering, training, supporting, and maintaining the product until users are satisfied. It is concluded by the product release milestone, which also concludes the cycle.



initial & evolution cycles



time duration for initial development cycles

e.g.:

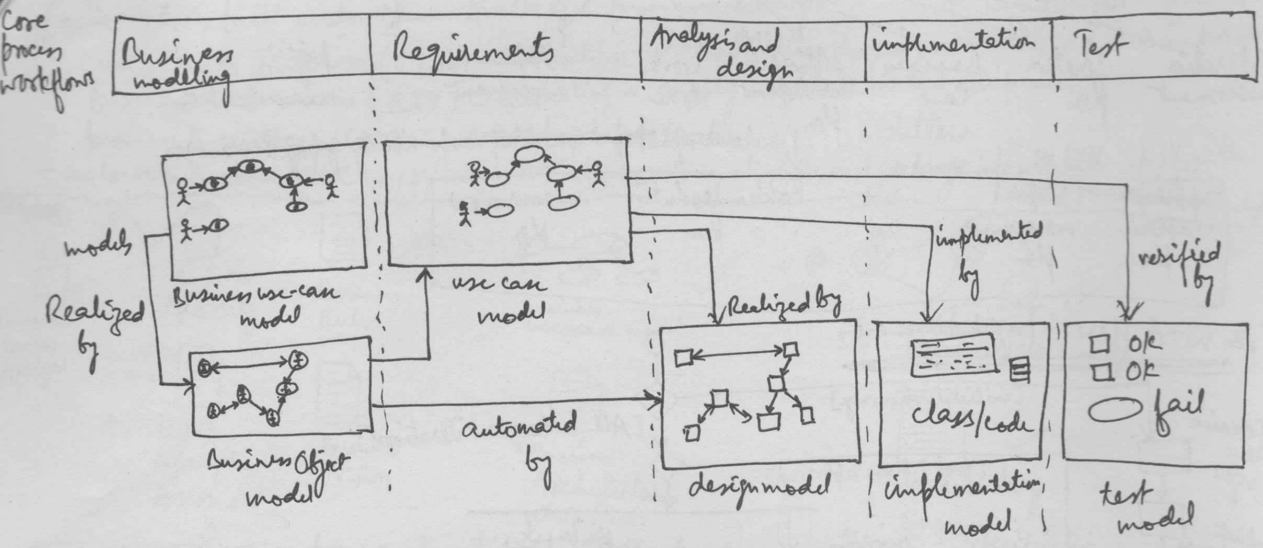
2 years project
- I = 2.5 ~~years~~ months
- E = 7 months
- C = 12 months
- T = 2.5 months

emphasis: inception phase— understanding the overall requirement and determining the scope of the development effort.

elaboration phase— focus on requirements, some software design and implementation at prototyping the architecture, mitigating certain technical risks by trying solutions, learning how to use certain tools and techniques. Finally, produce executable architectural prototype as the baseline

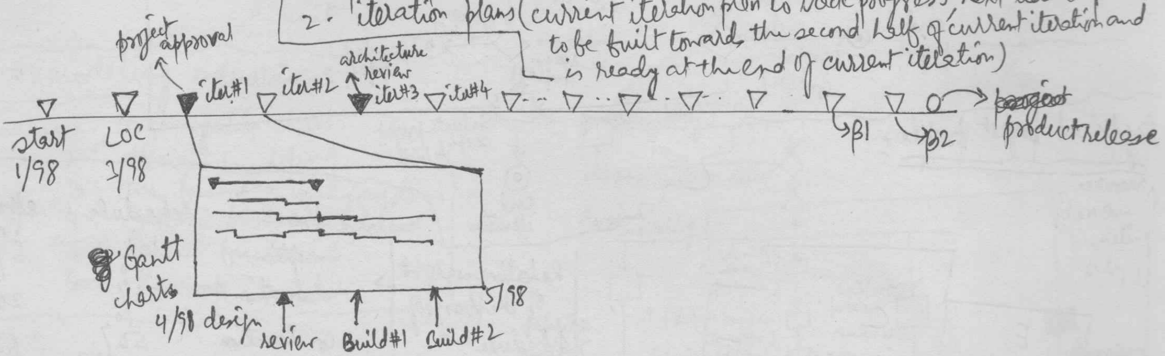
construction phase— design & implementation, first operational product

transition phase— ensure the system has right level of quality to meet objectives, fix bugs, train users, adjust features, add missing elements, produce final product use-cases "flows" through the various models —



Process workflows

* Project management workflow =
— planning an iterative project
✓ two levels of plans
1. phase plan (dates of major milestones, staffing profiles, dates of minor milestones)
2. iteration plans (current iteration plan to track progress, next iteration plan to be built toward the second half of current iteration and is ready at the end of current iteration)



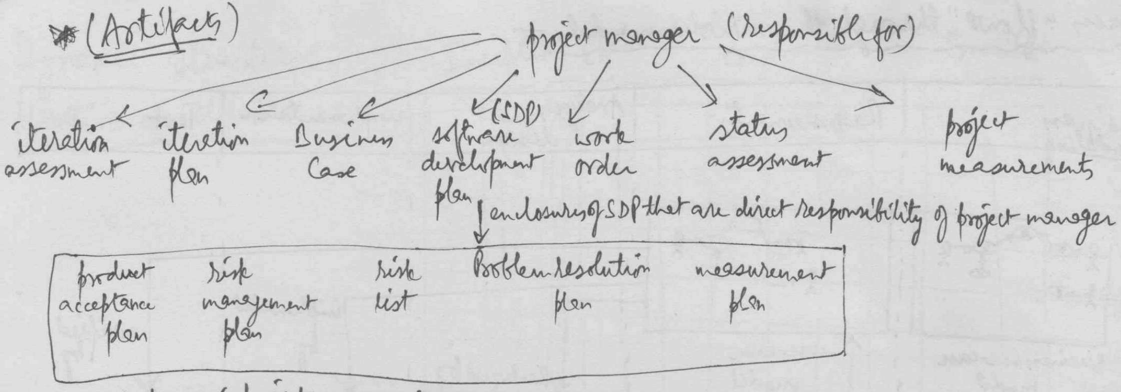
- Risk = know & ~~and~~ unknown potential problems
  - direct risk
  - indirect risk

* Risk has two attributes
  - the probability of occurrence
  - the impact on the project (severity)

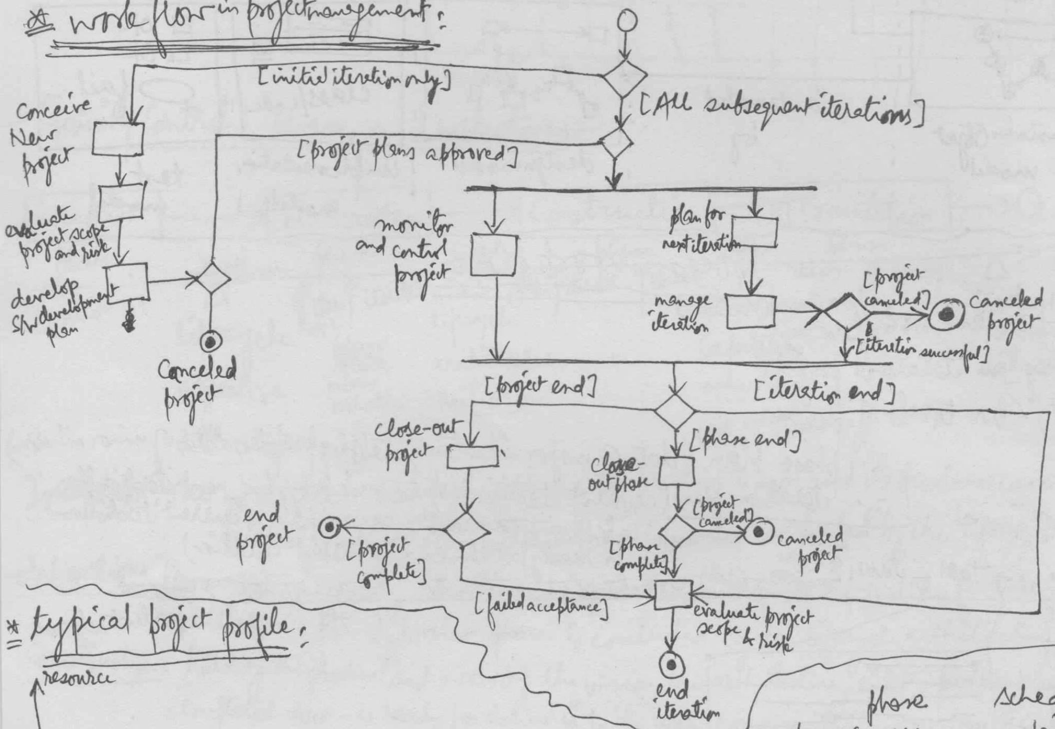
strategies: 1. risk avoidance
2. risk transfer
3. risk acceptance (mitigate the risk, define a contingency plan)

* metrics: - monitor progress relative to the plan
- improve customer satisfaction
- improve productivity
- improve predictability
- increase reuse

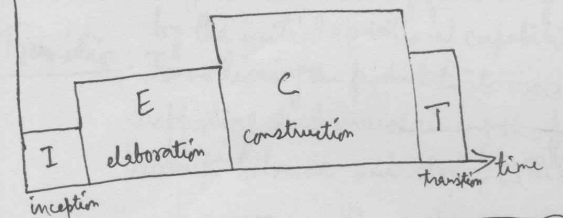
(Artifacts)



* work flow in project management:



* typical project profile:



Relative weight of the phases of schedule and effort for a typical project →

phase	schedule	effort
inception	10%	5%
elaboration	30%	20%
construction	50%	65%
transition	10%	10%

iteration duration for a range of iterative projects:- (example)

lines of code	number of people	duration of an iteration
5000	4	2 weeks
20,000	10	1 month
100 000	40	3 months
1000 000	150	8 months

number of iterations: [I, E, C, T]

low level = three: [0, 1, 1, 1]

~~high level~~ [1, 2, 2, 1]

typical level = six

high level = nine [1, 3, 3, 2]

"normal" projects have 6 ± 3 iterations

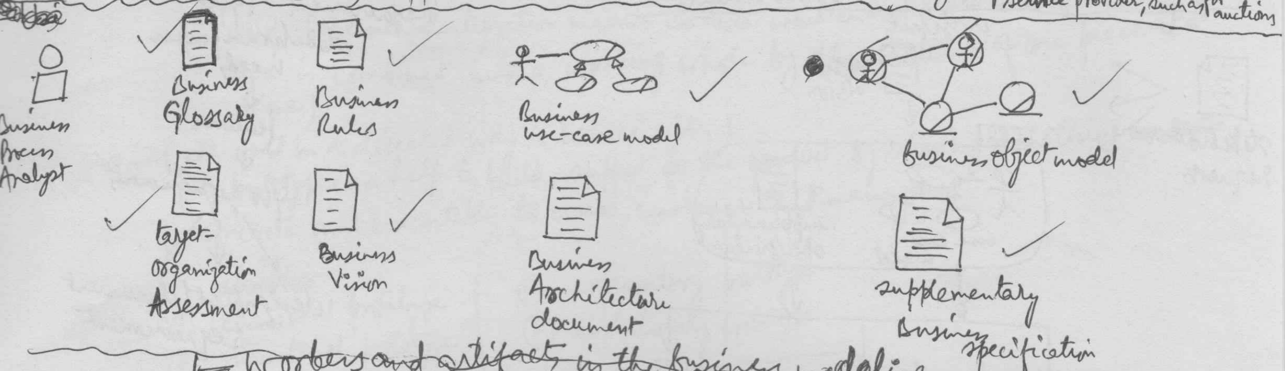
* building an iteration plan:

- define objective criteria for the success of the iteration
- identify the concrete, measurable artifacts that will need to be developed or updated and the activities that will be required to achieve this.
- beginning with a typical iteration work breakdown structure, massage it to take into account the actual activities that must take place
- use estimates to assign duration and effort to each activity, keeping all numbers within your resource budget.

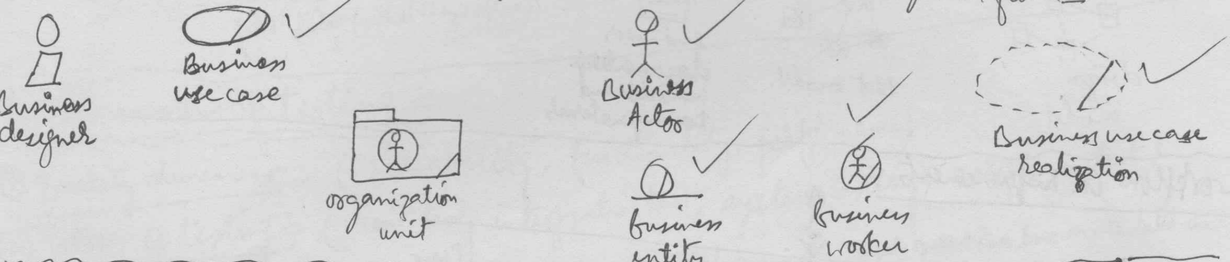
Business modeling workflow =

business tools built under the umbrelle of e-business development can be categorized as follows:

- customer to business (C2B) — applications that allow you to order goods over internet, e.g. electronic book stores
- business to business (B2B) — automating a supply chain across two companies
- business to customer (B2C) — providing info to customers, e.g.; newsletters
- customer to customer (C2C) — applications that allow customers to share and exchange info with little input from the service provider, such as for auctions

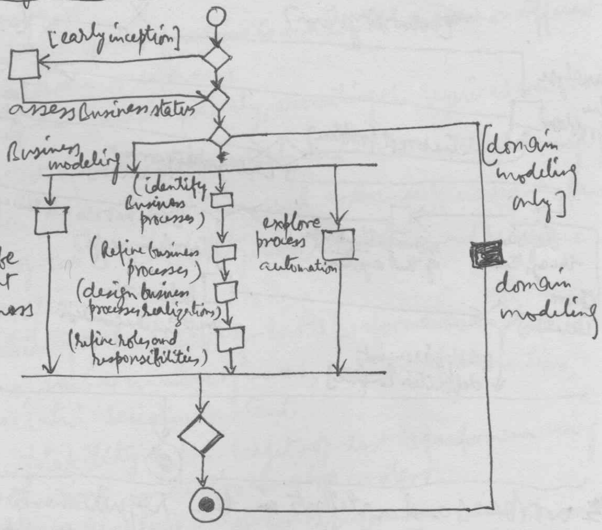


Workers and artifacts in the business modeling workflow =



Business modeling scenarios:

- scenario 1: organization chart
- " 2: domain modeling
- " 3: one business, many systems
- " 4: generic business models
- " 5: new business
- " 6: Revamp


[early inception]
assess Business status
Business modeling
(identify Business processes)
(Refine business processes)
(design business processes realization)
(Refine roles and responsibilities)
describe current business
explore process automation
[domain modeling only]
domain modeling

business modeling workflow

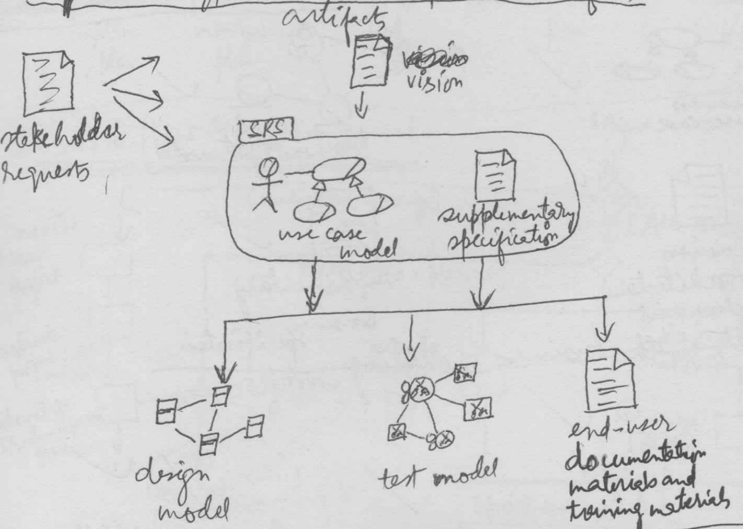
* the requirement workflow =

types of requirement : FURPS

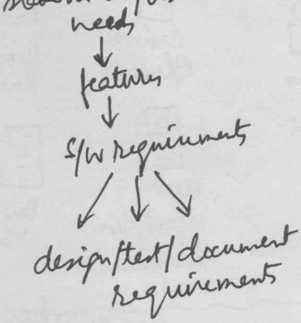
functionality usability Reliability performance supportability

(nonfunctional)

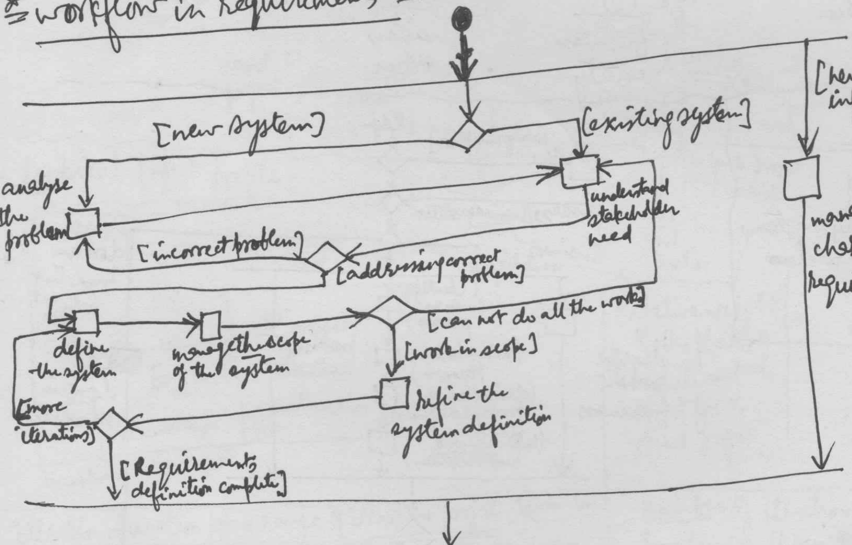
* Requirement types and relationships with artifacts =


artifact
vision
stakeholder requests
SRS
use case model
supplementary specification
design model
test model
end-user documentation materials and training materials

Requirement "Types"



* workflow in requirements =


[new system]
[existing system]
[new input]
analyse the problem
understand stakeholder need
manage changing requirements
[incorrect problem]
[addressing correct problem]
[can not do all the work]
[work in scope]
define the system
manage the scope of the system
Refine the system definition
[more iterations]
[Requirements definition complete]

* workers and artifact in the requirements workflow =

system analysts : Requirements management plan, stakeholder requests, glossary, vision, use-case model, supplementary specification, requirements attributes.

use case specifier : use case, use case package, software Requirements specification.

user-interface designer : actor (human), boundary class, user-~~of~~ interface prototype, use case storyboard

implementation workflow =

① builds (operational version of a system or part of a system that demonstrates a subset of the capabilities to be provided in the final product.)

during iterative S/W development there will be numerous builds.

② integration (a software development activity in which seperate S/w components are combined into a whole)
(Integration is done at several levels and stages of the implementation)
(In RUP, incremental integration means code is written and tested in small pieces and then is combined into a working whole by the addition of one piece at a time.)

③ Prototypes — (are used in a directed way to reduce risk.)
(A prototype can help to build support for the product by showing something ~~concrete~~ concrete and executable to users, customers and managers)

types: ~~behavior~~
① behavioral prototype
② structural prototype
first view (what they explore)

① exploratory prototype
② evolutionary prototype
second view (their outcome)

☆ dimensions of testing =

① quality dimension : (reliability, functionality, performance)

② stages of testing : (unit test, integration test, system testing, acceptance test)

③ types of tests : (① Benchmark test = compares the performance of a target-of-test to a known standard such as existing software or measurement(s)

② configuration test = verifies that the target-of-test functions is an acceptable manner on different configurations (hardware or software)

③ function test = verifies that the target-of-test functions properly, executing the required use case as intended.

④ installation test = verifies that the target-of-test installs properly and can be installed successfully on different configurations or under different conditions, such as insufficient disk space.

⑤ integrity test = verifies the target-of-test's reliability, robustness, and resistence to failure during execution.

⑥ load test = verifies the acceptability of the target-of-test's performance under varying operational conditions, such as number of users, number of transactions, and so on, while the configuration remains constant.

⑦ performance test = verifies the acceptability of the target-of-test's performance using various configurations while the operational conditions remain constant

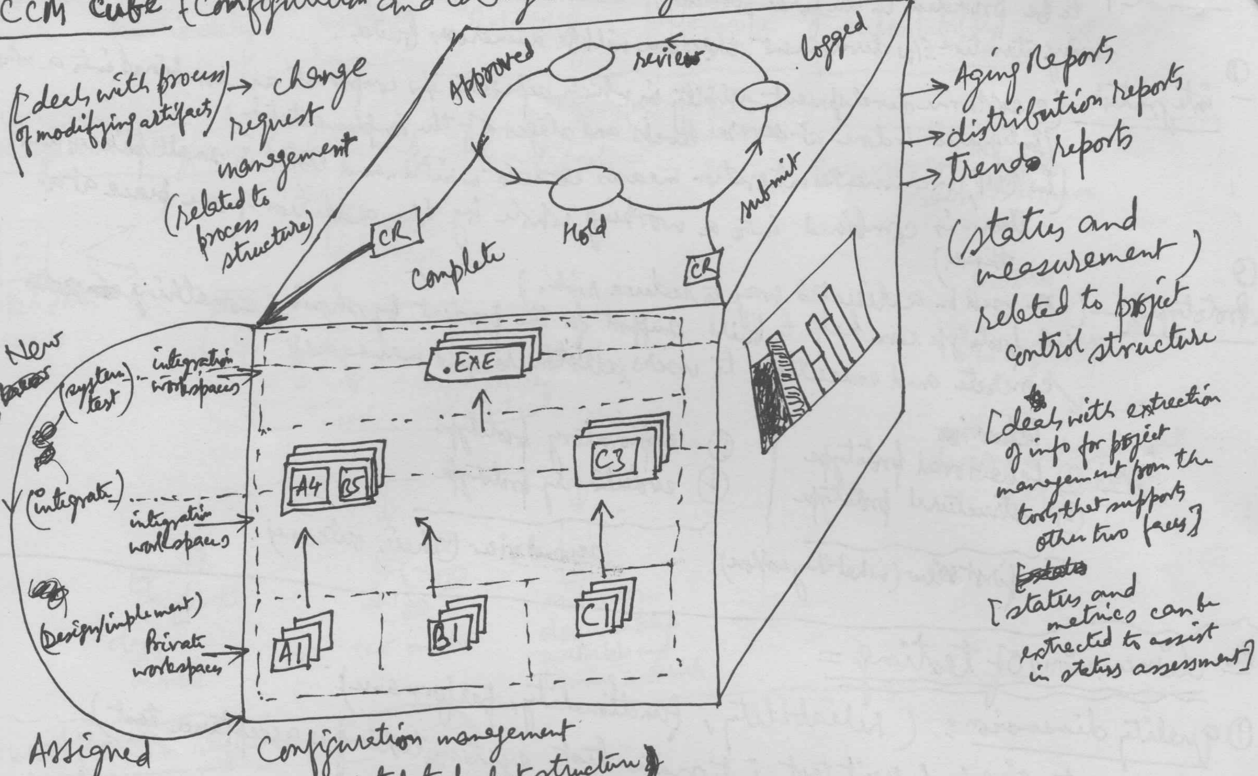
⑧ stress test = verifies the acceptability of the target-of-test's performance when abnormal or extreme conditions are encountered, such as diminished resources, or an extremely high number of users.

Regression testing : (a test strategy in which previously executed tests are re-executed against a new version of target-of-test, to ensure that the quality of target has not regressed -moved backward- while new capabilities have been added.)

The test model : ① test cases (set of test data, execution conditions, expected test results developed for specific test objective : Driven from use cases, design document, code)
② test procedures (set of detailed instructions for setup, execution, and evaluation of test results for test cases.)
③ test scripts (computer readable instructions that automate the execution of test procedures.)
test classes and components ;
test collaborations ;

* = Configuration and change management workflow =

CCM cube {Configuration and change management cube)



[deals with extraction of info for project management from the tools that supports other two faces]

~~[illegible]~~ [status and metrics can be extracted to assist in status assessment]

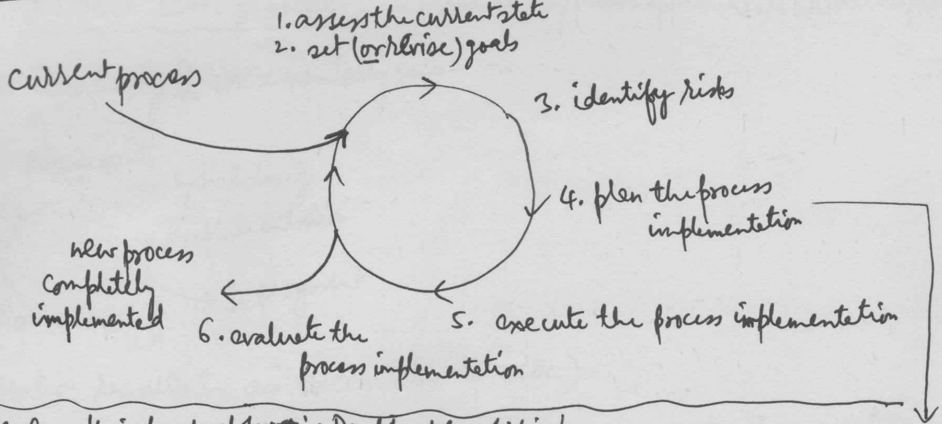
Configuration management
(related to product structure) ~~workspace~~
–[deals with issues of artifact, identification, versions, and dependencies among artifacts, identification of configurations, workspaces]

* = deployment workflows :

mode of deployment for ~~custom installation~~ (installable software)

1. in custom-built systems (by vendor) [may have associated custom-built hardware, i.e., software is required to run on specifically built target hardware.]
2. shrink-wrapped software (by user, delivered as a packaged product, i.e. install-wizards]
3. downloadable over the internet (by user, delivered over internet, i.e. setting up the product web site.)

Implementing RUP = (steps to implement a new S/w development process)

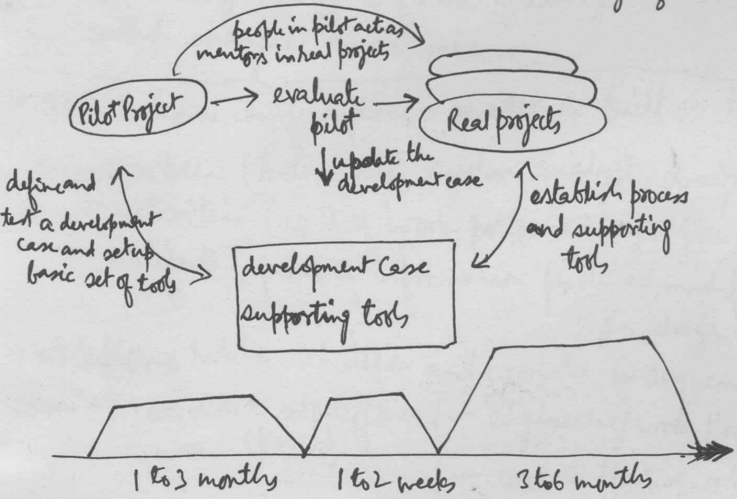

1. assess the current state
2. set (or revise) goals
current process
3. identify risks
4. plan the process implementation
5. execute the process implementation
6. evaluate the process implementation
new process completely implemented

* (~~RUP~~ adopting RUP usually involves develop~~ment~~ing Development Case, which's a project-specific version of process.)
* Development Case is likely to be a web site. It can be a modification of RUP online or a web site that refers to RUP online by way of hyperlinks

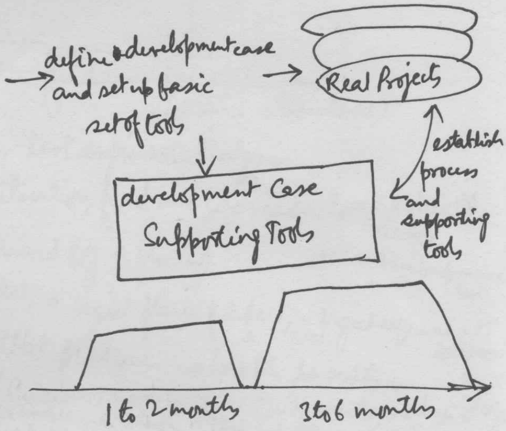
4. plan the process implementation

there are two approaches:

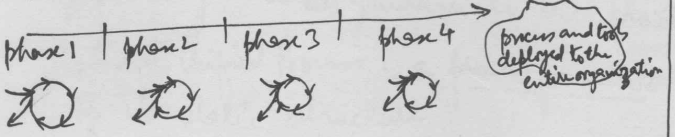
(i) "typical" approach — (trying part of RUP in Pilot Project before extending it to entire organization)

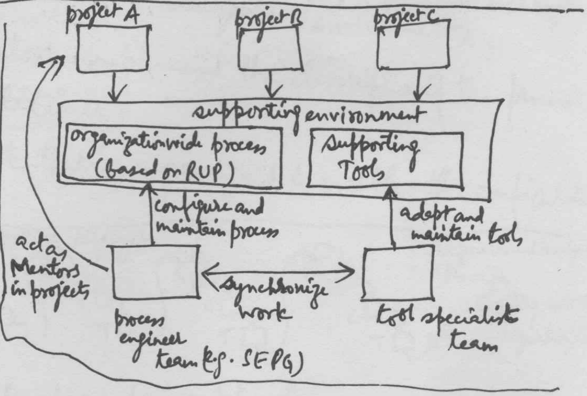

people in pilot act as mentors in real projects
Pilot Project
evaluate pilot
Real projects
update the development case
define and test a development case and set up basic set of tools
establish process and supporting tools
development Case
supporting tools
1 to 3 months
1 to 2 weeks
3 to 6 months

(ii) "fast" approach —


define development case and set up basic set of tools
Real Projects
development Case
Supporting Tools
establish process and supporting tools
1 to 2 months
3 to 6 months

* Implementing a process is a project —




project A
project B
project C
supporting environment
organization-wide process (based on RUP)
Supporting Tools
configure and maintain process
adapt and maintain tools
act as Mentors in projects
synchronize work
process engineer team (e.g. SEPG)
tool specialist team

Acronyms:
BPR = business-process reengineering
CBD = component-based development
CBT = computer-based training
CCB = Change control Board
CCM = configuration management, change management measurement
CM = Configuration management
CR = change request
CRM = " " management
PRA = project review authority
RFP = request for proposal
ROI = return on investment