Project Management

50367 characters in 6907 words on 1263 lines

Florian Moser

August 20, 2018

1 introduction

1.1 company

leadership, organization rules to manage specific task
setup for reoccurring, repeated task
skill-oriented architecture

1.2 project

temporal undertaking to create unique product/service
divided into work packages (which are then divided into tasks)
need different organizational structure

characteristics
performance specifications (form, fit, function)
known (bounded), unique solution for specific purpose
lifecycle with finite due date
resource requirements and tradeoffs
interdependencies & stakeholder conflicts

examples
pyramids, roman roads, cathedrals, castles (historic)
railroads, channels (gave rise to PM)
TESLA, apollo program, human genome project (successful)
london ambulance, airport berlin, diverse software (failed)

success factors
clear project goals, full management commitment
appropriate project organisation, professional project controlling
realistic planning of time/cost/tasks/resources/potential problems
explicit leader with vital interest in project success
qualified/motivated/adequate people

1.3 project livecycle

definition (low effort)
goals, specifications, scope
responsibilities, team

planning (middle effort)
work breakdown schedule (WBS), project plan
budgets, resources, risks, schedule

implementation (high effort)
status reports, quality audits
permute order of importance, react to new factors

delivery (middle effort)
train users, transfer documents
release resource, reassign staff
lessons learned

1.4 project management (PM)

science & art of solving a given problem within time & resource constraints
"shouldering just enough risk to escape with intact career"

importance
traditional hierarchical management declining
consensual management increasing
reliance on systems engineering increasing
make sure important projects don't fail

focus
delivering project as specified within budget & time limits

impact
better customer relations
shorter delivery times
lower cost / higher profit
higher quality / reliability / work morale

1.5 PM thinking models

waterfall
gather system requirements, then software requirements, then do analysis
program design, then coding, then testing, then operations

agile
establish vision, define product backlog, define release backlogs
choose elements for sprint backlog
execute sprint (ca 2weeks), holding daily short meetings (in SCRUM)
terminate sprint, reflect on changes & organisation, plan next sprint

1.6 PM problems

caused by management
wrong person chosen as pm
no responsible / no commitment / no interest
poor functional input in planing phase

caused by company culture
PM's role poorly understood
overcommitted resources in too many projects

caused by PM
poor control of function/non-functional requirement changes
no integrated planning & control
no project cost accounting ability
unrealistic planing, scheduling
conflicting project priorities
poorly organized project office

effects
late completion of activities
cost overruns & substandard performance
high turnover in project/functional staff
work completed multiple times

2 project management approaches

2.1 PMBOK

framework for PM principles
series of tools for PM and other stakeholders
no replacement for ISO 9000, SAMM

benefits
universal (accepted standards across multiple principles)
complete (the big picture of PM)
simple ("cliff notes" format to learn easily)

overview
10 knowledge areas help with process groups
process groups help with project livecycle

knowledge areas
stakeholders, integration, communication
resources, procurement (beschaffung)
scope, schedule, cost, quality, risk

process groups
initiation (defines, authorizes project phase)
planning (define objectives, plan actions to fulfil project phase)
executing (people & resources to execute plan)
monitoring/controlling (measure progress, corrective actions)
closing (formalize acceptance, halt the project)

project livecycle
starting
organizing & preparing
carrying out
ending

2.2 prince 2

project in controlled environment
1989 UK gov standard, open since 1996
revision 2009, free to use

principles
continued business justification (stop if lost)
learn from experience (draw lessons from previous work)
defined roles and responsibilities (clear organisation structure)
manage by stages (planning, monitoring & controlling in stages)
manage by exception (give authority to effectively work)
focus on products (clear delivery, quality, product definitions)
tailor for project environment (size, complexity, importance, risk)

themes
maintain record of business justification
define individual roles and responsibilities
define quality requirements & corresponding measures
create project plan, define which/how PRINCE2 is used
identify risks and opportunities
define how change will be handled
review viability and performance, decide whether to proceed or not

processes
directing (starting up, planning, closing)
planning (initiating part, delivery in bounded stages)

2.3 IPMA

2.3.1 ICB 3.0 (competence model)

technical competence range
fundamental PM elements ("solid elements")
project management success
interested parties, procurement & contract
requirements & objectives, scope & deliverables, resolution
risk & opportunity, quality, structure
resources, changes, cost & finance
organisation, teamwork, communication
control & reports, information & documentation
time & project phases, start-up & close-out

behavioural competence range
personal PM elements, attitudes & skills
leadership, assertiveness, engagement & motivation
openness, relaxation, creativity
results orientation, reliability, efficiency
consultation, negotiation, conflict & crisis
values appreciation, self-control, ethics

contextual competence range
pm elements related to project, managing inside organisation
project/program/portfolio orientation/implementation
permanent organisation
business, finance, legal, personnel managing
systems, product, technology
health, security, safety, environment

2.3.2 ICB 4.0

people
(required to succeed in projects)
self-reflection & self-management, personal integrity & reliability
leadership, teamwork, relations & engagement, communication
negotiation in conflict & crisis
result oriented, resourcefulness

project
(technical aspects or managing)
requirements & objectives, scope
organisation & information, stakeholder
risk & opportunity, design, quality
procurement, finance, resources, time
plan & control, change & transform, select & balance

perspective
(environment which must be navigated)
strategy, power & interest, culture & values
governance, structures, processes, compliance, standards & regulations

2.3.3 levels

higher include lower

level D
knowledge of technical elements

level C
application of technical elements
experience with limited complexity
might be given mentor to develop competence

level B
experience with complex situations
apply solutions within scope of project
experience with guiding other PM

level A
solutions align with permanent organisation
implemented technical elements/tooling/techniques

3 project instantiation

3.1 general process

collect project ideas, generate new ones with creative activities
select a few to perform pre-study
create initial project definition, put it in project portfolio
do external partner search, sign contracts if found
create project charter, define & clarify objectives
start project planning

3.2 pre-study

align project with strategy, existing portfolio
define project framework (ICB, prince 2, ...)
identify stakeholder needs/requirements
evaluate technical/economical feasibility
conduct initial risk analysis

3.3 project portfolio

pool of projects; choose the right ones to pursue
successful firms choose strategic over purely financial projects

identify strategic projects
support multiple goals at the same time
direct organisational improvement
enhance, enable key areas

prioritize potential projects
limit active projects
identify risk-intensive efforts
balance short/medium/long term returns
no projects via backdoors

3.4 project selection models

criteria
realism
capabilities (simulate decision effects)
flexibility (results within range of conditions)
ease of use (convenient, executable, understood)
cost (low modelling costs relative to size)
easy computerization (easy data handling)

non-numeric model
sacred cow (suggested by superior)
operating necessity (required to keep operational)
competitive necessity (required for competitive position)
product line extension (extend/strengthen/replace existing)
comparative benefits (select by largest benefit)

numeric model
payback period (till investments payed back)
average rate of return (profit average annually)
discounted cash flow (money now > money later)
internal rate of return (hold present profitability)
profitability index (highest percentage of payoff)

3.5 project charter

written document, between client & pm

tasks of client
decision on pm & organization
responsible for regulations
project phase & milestones
project priorities
support of pm

example

client, pm
problem, purpose, scope
cost, start, end
external factors, interdependencies, deliverables, risks

3.6 project planning

systematically gather information & predict future activities
define purpose & vision of project
predict project outcomes required to fulfil vision
define project scope, objectives, targets, deliverables
structure tasks & processes to create schedule
define target measures (cost, quality, team, results)

goals
explain key benefits
team, capacities, info determine schedule & quality
team, schedule determine profit (cost vs benefits)
"improve overall performance" as global goal
"improve db connection" as classes of goals
"minimize db latency" as operationalized goal

MOSCOV functionality analysis
Must (critical functionality)
Should (if all must can be fulfilled)
Could (can if all more important features can be fulfilled)
Wont (not this time, but remembered for future projects)

3.7 PM control

monitor project to ensure target measures can be fulfilled

tasks
fix inter-personal, capacity, requirement problems
inform stackholders according to interest & influence

4 structuring a project

4.1 create work packages

do jointly with executing team

work breakdown structure (WBS)
establish project culture
divide work into hierarchical units
tree structure or table

function oriented WBS
focus on task groups
design, manufacturing, distribution
level 1 project title ("bike")
level 2 activities ("concept", "manufacturing")
level 3/4 tasks/subtasks ("build", "test")

object oriented WBS
focus on system hierarchy
electrical, mechanical system
level 1 system name ("bike")
level 2 part name ("frame")
level 3 smaller parts name ("screws")

process/phase oriented WBS
focus on project phases
planning, implementation
level 1 process name
level 2 general work package (mop, vacuum)
level 3/4 detailed work plan

work package specification
unique logical identifier
define scope, deliverables & how performance is measured
less than 80h, shorter than report period

tips for creation
integrate diverse team members
design multiple WBS, decide in second meeting
focus on tasks, not responsibilities
divide in 2-8 work packages
consistent measurements, terminology
after definition, map packages to business units

4.2 structure packages

logically sequence work packages (dependencies)
create components consisting of sequences
integrate components in plan
use task list for small, use graph for big projects
estimate capacity, schedule, cost, investment for each work package

estimation methods
analogy (similar project already completed)
relation (structure comparison)
multiplication (expendidure per unit method)
parametric (with established parameters)

4.3 capacity planning

identify capacity per work package, sum up
compare with availabilities, adjust if needed
ensure quality/quantity of resources available

challenges
missing goal definition, scope changes
interdependencies with scheduling
difficult to quantify

4.4 scheduling

WBS produces a work flow of all packages
integrate work flow in schedule
depending on method need additional meta info besides package list

schedule list
need start or end date of activities
simply list all activities, possibly assigning responsible, reports, ...
simple, fast, no special requirements
but no interdependencies visible, hard to keep track

gantt charts
need duration, chronological order of activities
graphical description of schedule, tasks as rows, time as y axis
activities denoted as bars, with rich information (use appropriate software)
denote milestones, visualize interdependencies (but only limited possible)
visualize importance (colours), milestones (red squares), groups (bars)
common, self explanatory, easy to create, shows parallel activities
but hard to change, large planning periods needed for clarity

network
need duration, interdependencies of activities
create workflows, denoting both logical, chronological interdependencies
visualize importance (form/border-thickness of boxes)
can align employees, costs to different tasks, multiple evaluation methods
but rather complex to use, difficult to change, training needed
activity on arrow (AOA) for waterfall projects
activity on node (AON) for parallel transactions
event on node (EON) for event-determined progress
node information includes earliest/latest start/end, duration, float
node contains all info of activity (possibly connected to WBS)
deterministic duration estimation by calculating critical path
can use PERT (stochastic method) for probabilistic estimation
for optimistic a, realistic m, pessimistic duration b
expected duration = ((a + 4m + b) / 6); variance (b-a/6)ˆ2
but hard to connect to timeline, distinguishing activities hard

schedule compression
overtime usage (but more flaws, slower progress)
more ressources (but aquire needs time, existing ressources do training)
reducing scope (but customer needs all agreed on features)
outsourcing (but insecurity with quality/schedule of supplier)
overlap serial activities (but increased risk/rework of changes)

4.5 cost planning

calculation terms
cost calculation (effective consumption / claim of production factors)
cost-type accounting (recoding of cost at occurrence; material, HR)
cost-centre accounting (cost-type (people) to cost-centre (HR) in period)
cost-unit accounting (per produced good, delivered service)

principles
top-down (fast & easy; using parameters, key data)
bottom-up (for detailed cost planning, project controlling)

example
HR (developer, in CHF/h, 130/h)
material (steel, in mˆ2, 20/mˆ2)

project calculation
sum of all direct work package cost
∗1.1 overhead cost (office, management, infrastructure)

∗1.5 risk/profit margin
results in offer price

cost estimation methods
single expert (educated guess based on experience)
classical delphi method (multiple experts, no discussion)
broadband delphi method (multiple experts, discussion)

5 project organization

needs intensive interdisciplinary cooperation
needs specific tasks, workflows, leadership tasks & structure, motivation

5.1 internal organization

5.1.1 tasks

result, fact oriented arrangement of functional structures
clear tasks, responsibilities, competencies
effective collaboration & cooperation of all parties
fast alignment of organization to changing objectives & conditions

5.1.2 people

expert committee & steering committee advise project leader
project leader controls special functions, project team

5.1.2.1 steering committee

temporary till end of project, to be reported to by PM & team
summary of decision makers/responsibilities (users, business)
take essentials decisions together

tasks
represent interests of participating divisions, outsiders, related topics
assign persons to consulting team
define project objectives, tasks & priorities (with PM)
control & approve project plan
verify & approve phase results
offer support, advice, mediation for major problems
take decisions superseeding competency of PM
in charge of resource allocation

composition of representatives
project manager
all relevant hierarchy levels
all essential involved parties
experts
possibly external groups ("political representatives")

representatives criteria
in a position to enhance implementation or involvement
are expertise or information carrier

5.1.2.2 expert committee

inform, advice, support project team
expertise for effects & risk
carry out results of project into specialized fields
but no decision-making authority

5.1.2.3 project manager

needs to reach project goal in terms of content & competence
assures best possible technical & economical success of project

needs (must object if not provided)
clear project brief
realizable, agreed on & measurable objectives
well-defined area of responsibilities
sufficient resources
congruent tasks & authorities

tasks
align tasks, competencies, responsibilities, resources, priorities
push the project, start initiatives
realize project within given scope (activity, cost, deadlines)
initiate contacts & assure communication
lead & motivate project team
prepare & procure decisions, clear up conflicts
assure & control system & project tasks

5.1.2.4 special functions

consulting
documentation management
connected to the concerned parties
power/technical/social promoter

5.2 external organization

5.2.1 choose appropriate structure

determining factors
structure of organization, experience with projects
size, duration, risk, frequency, number of project
importance to business, availability of resources
necessity for interdisciplinary cooperation

need for a change
due to natural workflow of project (changing content, objectives)
because important staff members change

5.2.2 pure project organisation

for extraordinary project (major size, importance, uncommon subject)
board of management controls projects directly (same level as departments)

PM
manages separate organisation for project
direct superior of team members

pro
fulfilment of project goals
high identification with project
full line authentication of project manager, clear authority
fast reaction to changes, difficulties

cons
allocation of dedicated resources
little exchange with specialists in divisions
reintegration of team members after end of project

why
of high importance for business

5.2.3 influence project organisation

for small projects
team members continue to do their main line tasks
functional organization & hierarchy remain unchanged

PM
directly above departments, executive management function
information & advising functionality, no decision authority
tracks project content, deadlines, cost
motivates employees within departments
department manager has to accept PM

pro
work force may be part of multiple projects (can combine experience)
no change of company organization, no break from department

contra
limited identification, responsibilities
limited across department problem solving
response, decision time low

why
creativity, brainstorming
uncertain performance

5.2.4 matrix project organisation

for efficient, profitable projects if organization fulfils requirements
combination of vertical (departments) and horizontal (projects)
departments act as service providers for projects
leverage knowledge & expertise from past projects
needs flexible, qualified employees with understanding of leadership

PM
widely accepted & political
responsible for whole project
may shares resources with other PMs
team members jointly supervised with department

pro
project dimension, professional dimension balanced
goal oriented coordination of different interests
high identification of all employees with the project
simultaneous technical source for all team members

cons
conflicting interests
cut of authority (competencies arguments, passive resistance)
enormous coordination effort between departments & project management
difficult for employees (two masters)

why

high interdisciplinary skills needed

5.3 risk management

risk vs projects livecycle (early is high risk but cheap, and vice versa)

risk matrix
probability (frequent → never) vs impact (high → none)
protect combinations of probability & impact
take effect related actions to reduce impact
take cause related actions to reduce probability

opportunity matrix
probability (frequent → never) vs benefit (high → none)
often & high benefit opportunities for financial / know how gain
rare & low benefit for networking / image gain

opportunity-risk profile
visualize risk & opportunity matrix at the same time
see difference between two points, draw line
lower values in risk dimension are better

risk process
define focus when initiating process
identify, evaluate risk with an analysis
avoid, reduce, transfer, share, take risk
control risk upon execution

5.4 monitoring

collection/recording/reporting of information concerning performance
type of project determines details of monitoring

information flow
engineers & developers collect
administration reviews
management approves, may takes corrective actions
similar flow in both planning/scheduling and reporting/monitoring

cycle
closed continuous cycle with planning, monitoring, controlling
revised plans & schedules are followed by corrective actions

prerequisites
realistic & complete planning data
actual data corresponding with planned data

phases
gather realistic data
variance calculation, analysis & evaluation
actions undertaken, plan revised

possible reasons for variance
bad planning (complexity underestimated, scope creep, no historic data)
process changed (new information, alternatives, priorities)
supply changed (insufficient qualifications, missing capacity)
mistakes were made (insufficient quality, inefficiency)

design monitoring system
identify key factors to be controlled
PM must define precisely which characteristics to keep track of
specify exact time, cost, performance boundaries
establish procedures, criteria, standards for all measured factors
may collect customer, operation, engineering, specification data

data collection reports
generate progress reports after data collection
include project status, time/cost, variance
identify cause/effect, note trends, visualize recent data

5.5 controlling

primarily by changing assets, altering activities
PM must pay attention to both conservation & regulation
PM must guard physical, human, financial resources

5.5.1 underlying issues of problems

performance
technical, quality, reliability, interfunctional problems
insufficient resources available
client requests changes in specification

cost
price/quantity of resources changed
scope increases, bids too low
reporting poor/inadequate
corrective control was too late

time
time estimates too optimistic
incorrect task sequencing, preceeding tasks incomplete
required resources unavailable when needed
regulations, specifications changed

5.5.2 types of control

physical assets
preventive/corrective maintenance of assets, inventory
set timing (minimize faults & interference), quality of maintenance

human resources
provide fertile ground for cultivating people

5.5.3 processes

go/no-go control
check for preconditions, execute if exists

post control
solving issues which caused problems

cybernetic control
self-correcting feedback loop
(1) project baseline (agreed upon, accurate)
(2) project work (means to measure actual work)
(3) variance identification of (1) and (2)
(4) variance correction

5.5.4 controlling change

single most important problem
changes due to customers/members to improve end result
PM must control the way changes are introduced

change control system
review all requested changes and identify impact
translate impact to performance/schedule/cost
evaluate cost/benefit of changes
accept/reject changes and inform concerned parties

change order
describing a change
defining deliverables & impact to cost, schedule, performance

procedure
project contracts define change order processing
any change must be in the form of a change order
changes must be approved by customer & executioner
PM must be consulted when creating the change order
if change order is approved, integrate in master plan

5.6 monitoring & control systems

if discrepancy noticed hold meeting, borrow resources, notify client

relative cost & time variance
grid with variance in cost / variance in schedule
draw projects as cycles (size for budget, fill for completion percentage)
add arrow with estimated tendency

earned value chart
aggregated performance measure, reports cost & performance
take into account amount of work completed relative to cost
multiply completion percentage with planned cost for each task
draw planned baseline, actual cost, actual earned value
read out time / cost (y-axis), schedule (x axis) variance
cost C, schedule S, time T performance index by dividing key factors
ratios helpful to compare to other projects

5.7 leadership

5.7.1 locomotion

motivation, activation of employees
fulfilment of goals

5.7.2 cohesion

collaboration, loyalty in the group

5.7.3 leadership theories

character theory
leaders have certain traits which produce consistent behaviour pattern
are be born with this, may develop over time, but can't be taught
skills in communication, decision making, integrity, empathy
but scarce evidence saying height, self-confidence, extroversion

behavioural theory
investigates how leaders behave
multiple behavioural traits possible; can be learned or taught
autocratic (does not consult team, good for fast decisions)
democratic (involves team & seeks consensus, good for team work)
laissez-faire (lets team take decisions, good for mature teams)

situational theory
postulates leadership should depend on environment
ask if least-preferred coworker friendly, cooperative, supportive, open
answer score then indicates if task or relationship oriented leader
if score high then relationship oriented, good for favorable conditions
if score low then task oriented, works even in unfavorable conditions

interaction theory
investigates influence of interpersonal interaction to leadership
assumes that humans have complex, changing motivations
open workplace, dynamic exchange between leader & follower
depending on personality, values, group, norms
if relationship oriented then value respect/trust/communication
if task oriented then value actions/productivity boosts

5.7.4 leadership models

path-goal
archievement oriented leaderships (high goals, performance, confidence)
directive leadership (direct, clear instructions, advice)
participate leadership (team involved in decision making)
supportive leadership (show personal concern for well being)

tannenbaum/schmidt
manager shares power with non-managers
manager makes temporary decision & sells it, answers questions
non-managers provides input, make limited/joint decision

blake/mouton
concern for people vs concern for production
country club style (most concern for people, happy people work hard)
impoverished leader (no concern for people or production, bad)
middle of the read (average concern for both, average performance)
team leader (high concern for both, very good performance)
produce or perish (only concern for production, needs punishments)

hersey/blanchard
relationship vs task focused behaviour (situational theory)
task focus depends on maturity (skill) of followers
mature followers need low task focused behaviour
relationship focus depending on confidence (motivation) of followers
confident followers need low relationship focused behaviour
delegating for experienced, willing employees (not much to do)
participating for experienced, unwilling employees (offer advice)
selling for inexperienced, willing employees (offer advice & directions)
telling for inexperienced, unwilling employees (explicit directions)

5.8 conflicts

origins
distribution, evaluation, personality, goals (uncontrollable)
roles, perception, relationship (controllable)

livecycle
forming (all nice & easy)
storming (conflicts arise, roles are created)
norming (common behaviour established)
performing (most productive phase)
ending (group goes own ways again)
start most issues concerning schedule, technical
end most issues concerning schedule

deal with conflicts
avoidance, circumvention, escape, delegation
compromise, fight, negotiation

conflict escalation levels
win-win (hardening, debates & polemics, actions not words)
win-lose (loss of image, loss of face, strategy of threads)
lose-lose (limited destruction blows, fragmentation, together in the abyss)

5.9 negotiation

needs encapsulation by interests, encapsulated by positions

principled negotiation
conflict resolution technique aiming for win-win result
(1) be hard on problem, soft on people (keep separate)
(2) focus on interests, not positions (concerns, hopes, interests, fears)
(3) invent options for mutual gains (win-wins)
(4) objective criteria for evaluation of claims, options, alternatives

other important points
active listening, show interest, appreciate interests of the other party
good negotiation climate (demanding discussions, food for thoughts)

process (soft to hard)
contact/communication (show problem, agree on problem)
influence (understand interests, then show solution)
contractual variables (work on alternatives based on interests)
negotiation (time pressure, have BATNA)

BATNA
best alternative to no agreement, can be done without other party
fall back position (this happens if negotiations fail)

5.10 auditing

formal inquiry about project aspects
focuses on what management desires
evaluation must have credibility with management & PM

goals
translate archievements into contribution to company
understand project strengths & weaknesses

dimensions of success
efficiency of project (budget & schedule)
customer image/satisfaction (most complex dimension)
business success (money)
future potential (nebulous)

content
current, future status of projects
status of crucial tasks
risk assessment
information crucial to other projects
limitations of report

comments on
methodology, procedure, records, properties
budgets, expenditures, degree of completion

result
far broader than simple financial audit
get set of recommendations for future usage

5.11 termination

types
extinction (completed, failed, changes in environment)
addition (becoming part of organisation)
integration (assets absorbed by parent)
starvation (no more budget)

areas
organization (plans, personnel)
financial (payables, deliverables, budget report)
purchasing (contracts, supplier communication, final payments)
site (close facilities, dispose equipment)

final report
historical recap
project bibliography (good, bad, ugly)
documentation (project plan, change order, audits)
performance of project & administration
performance of organizational structure & project management
recommendations for the performance reports
lesions learned (benchmarks, killers)
to enable better project management in the future

6 agile principles

6.1 complex systems

6.1.1 complicated vs complex

complicated projects have little insecurity (go to the moon)
complex projects have great uncertainty (raising a child)

6.1.2 problem types

simple (few, known variables)
organized complexity (moderate amount of unrelated variables)
disorganized complexity (like understanding laws of nature)

6.1.3 agents

all components of the application
interact/influence each other in unpredictable ways
regularities emerge, form patterns which feed back into system

6.1.4 complex system elements

open systems where agents interact in an interesting way
small disturbance lead to very different outcomes (non-linear)
ability of self-organization, self-stabilization, information processing

general workflow
numerous, simple, self-organized local relationships emerge
complex adaptive behaviours amplify or damper the local relationships
all under changing external environment, info flows in / outside

emergence
seemingly random interactions dictate the behaviour as a whole
disequilibrium (economical like market changes, physical like heating)
symmetry breaking (economical like business process reengineering)
experimentation (enforce different configurations like team building)
reconfiguration (try new parameters in new environment)

self organization
increased organizing (still within limits of initial configuration)
tension & threshold (work goes beyond system limits)
newly emerging configuration (critical event initiates new configuration)

path dependence
past well known (real state / company history)
but future unsafe

lack of predictability
influence of random noise
effect of environment
lack of knowledge of initial system

6.2 agile project management

guide vision because non-material fields continuously influence behaviour
see members as intelligent, skilled professionals as basis for teamwork
establish simple rules to support complex, overlaying team behaviours
promote open information to allow team members to react to changes
allow restricted autonomy ("light touch") to intelligently control
be vigilante in leading; continuously monitor, learn & adapt

6.3 agile manifesto

values
individuals & interactions over processes & tools
working product over comprehensive documentation
customer collaboration over contract negotiation
responding to change over following a plan

principles
satisfy customer with early & continuous delivery
welcome changing requirements for competitive advantage
deliver software frequently
business and development collaborate
trust motivated individuals to get the job done
face-to-face communication for efficient conveying of info
working software as principal measure of success
ensure all stakeholders can maintain pace indefinitely
give attention to technical excellence & good design
maximize "work not done" (simplicity)
self-organise to ensure best requirements, architectures & designs
reflect regularly on processes and tune behaviours

6.4 agile methods

lightweight, people based rather than plan based
several methods (XP, scrum, rapid results, ASD, DSDM)
agile manifesto closest to definition

workflow
product backlog fuels sprint backlog
sprints in 2-4 weeks, daily scrum every 24h
ends with a potentially shippable increment
sprint review ends sprint and marks start of next sprint

fundamentals
team, version control, iterative, incremental development

teams
sustainable pace, team
project charter (high level summary of objectives)
scrum of scrums (for scalability, ambassadors of daily scrums meet)
team room (dedicated room for team with any needed material)
heartbeat retrospective (meeting to improve work quality)
facilitation (fruitful meetings lead by facilitator)

product management
backlog grooming, user stories, incremental development
personas (detailed, synthetic user descriptions)
story mapping (grid of priority, complexity of user stories)
story splitting (split large user stories, still business value)
3C's (card → conversation → confirmation; feature workflow)
INVEST (independent, negotiable, valuable, estimate, small, testable)

design
refactoring, simple design
CRC (class, responsible, collaborator; RPG to test design idea)
quick design session (compare design alternatives in pairs)
rules of simplicity (unit test, no duplication, seperated concerns, minimal)
ubiquitous language (use vocabulary of given business domain)

lean
definition of done
lead time (requirement to fulfilment time)
kanban board (task board with limited items per columns)
kanban is continuous (not based on sprints)

scrum
iterative development, iterations, daily meeting
timebox (work on task till time done, then evaluate)
three questions (done, to do, blocked?)
burndown chart (show story points completed / remaining)
task board (todo, in progress, done; sticky notes)
definition of done (requirements for user story to be done)
definition of ready (requirements for user story to start)
point estimates (story points; assign complexity to user story)
relative estimation (group items of similar complexity)
planning pocker (hidden choose cards, low&high explain reasoning)
backlog (necessary & sufficient todos to complete project)
backlog grooming (review backlog with product owner)

extreme programming
daily meeting, iterations, frequent releases
pair programming (driver & navigator, switch often)
sustainable pace (no overworking)
sign up (choose tasks to work on)
velocity (give points to user stories, count how many completed)
user stories (defines what in short phrase, implemented in sprint)
collective ownership (nobody owns code, anyone may modifies any file)
continuous integration (unit tests upon git push)
simple design (justify pattern cost, YAGNI)
refactoring (restructure code to make it more robust, easier)
TDD (test driven development, first failing unit test, then implement)

testing
usability testing (end user fulfils given task)
exploratory testing (continuous testing by autonomous tester)
unit tests (pass/fail execution of short part of code)
TDD (single test → write just enough code to pass → simplify)
mock objects (mock implementations)
acceptance tests ([automatic] user story test)
ATDD (acceptance test driving development, customer creates)
ATDD answers what problem, how solved, what about edge cases
BDD (behaviour driven development, synthesis of TDD, ATDD)
BDD answers five Why's, defines task cleary, focuses on business value
given-when-then (template for acceptance tests for user stories)
role-feature-reason (to create user stories; "as a ... i want ... such that ...")

devOps
version control, automated build, continuous integration & deployment

6.5 SCRUM

"agile process delivering highest business value in shortest time"
inspect actual working software in short, regular intervals
business sets priorities, but team self-manages

characteristics
self-organizing teams, no given specific engineering practice
generative rules to create agile environment
product progresses in monthly sprints
requirements captured in backlog (list of pending features)

workflow
from product backlog, a sprint log is created
daily scrum meeting during the sprint
at the end of sprint shippable product increment available

roles
product owner (defines & manages product backlog, prioritizes)
scrum master (ensures agile values, principles)
team (shared accountability)

sprint
full RE, design, coding, testing during single sprint
no changes during sprint, but scope negotiations with product owner

sprint planning
meeting at start of sprint
select items from backlog for next sprint log
whole SCRUM team participates

sprint review
informal meeting at end of sprint
set next objectives to increase value
key stakeholders invited by product owner
product owner explains what backlog items have been Done & not Done
development team discusses good, bad, found solutions, demonstrates Done
product owner discusses state of project, estimates delivery dates
all collaborate what to do next as input for sprint planning
review if value propositions still hold
review timeline, budget, potential capabilities
backlog adjusted to reflect results of sprint review

sprint retrospective
after sprint review, before sprint planning
focus in inspection, adaptations to improve processes
answer what went well, what did not, how to improve in next sprint

daily scrum meeting
team reflects on work from past 24 hours (done)
team plans work for the next 24 hours (todo)
team resolves interdependencies issues (blocked?)
yesterday done, today do, any blockades?
for self-organization, honest status, focused on outcome (not progress)

scrum of scrums
ambassador of scrum goes to bigger meeting
frequency of meetings based on coupling
include up to 800 people in single scrum process

product backlog
single source of requirements for product finish, constantly adapts
features, functions, requirements, enhancements for next iteration
continuously refine items till able to do in one sprint
use backlog to estimate development speed, release dates, etc

sprint backlog
selected items from backlog, at least one hig-level, high-priority one
can only be modified by development team during the sprint
keep track of work remaining, work to do and put it in chart

burndown chart
publicly displayed chart visualizing current work status
remaining effort (% completed), remaining tasks (done, not done)
ideal burndown (line), completed tasks (as boxes)

pros
fully developed, tested features in short time frames
simple process with clear rules
increased productivity, improved communication
self organization, each team member has responsibility
can be combined with extreme programming

cons
undiciplined hacking with no real documentation
violation of responsibilities

6.6 extreme programming

"extremifies good programming practices"
code inspections are good, therefore code in pairs (inspect all the time)
testing is good, therefore write tests before code, run repeatedly
customer contact good, therefore have representative in team

activities
coding (code is most important tool, use for communication)
testing (unit test for system, acceptance tests for customer)
listening (give feedback to customer business logic)
designing (create design structure, decompose units)

principles / values
communication (dev's & users have shared view of system)
simplicity (code for today, YAGNI)
feedback (unit tests, acceptance tests, planning pocker)
courage (code for today, refactoring, removing old code, persistence)
respect (self & others, don't waste time of others)

key ideas
code in pairs
stay in contact with the customer
create tests before coding, afterwards test heavily
short iterations
KISS, YAGNI
collective ownership

why controversial
no specialization (analysis, architecture, coding, testing done by all)
continuously analyse & redefine
develop framework & infrastructure on the go, focus on business value
dont maintain documentation (face-to-face, good code)

6.7 rapid results

"drive long term strategy with short-team rapid results "
challenge team to archive measurable results in short time
integrate all activities rather than department focus

rapid result focus
activity statements → concrete results
"exercise regularly" → "get 2kg muscles in 2 weeks"
strategy → set of rapid results
"produce new product" → "sell product to first client in 100 days"

benefits
immediate results and sense of victory
learning of new skills by practising them
greater confidence to do research & publish

attributes
ownership & excitement, experimentation, collaboration
focus, follow through, output orientation

crisis environment
helps to make innovation happen fast
people work as a team, ignore red tape, question assumptions
people can transcend functional & hierarchical boundaries

design rapid results using crisis factors
short timeframe & clear measure of success (success near & clear)
stretch goal only archievable with innovation (challenge)
urgent & compelling area to work on (sense of urgency)
experimental, fun, innovative (exciting, game-like)
visible leadership demanding measurable results (high stakes)

low-hanging fruit (supposedly boring)
handle simple items to get quick results
focus on easy stuff, no learning required, no maintenance required

rapid results (supposedly fun)
scope out result with large improvement opportunity
focus on specific area with both hard & easy problems
learnings as important result
change work processes to sustain results

milestones project
consider milestone, determine how much work can be done till then
analyze possible improvements, execute ones which are most promising

100 day results project
try to fit all milestones in 100 days
probably need to ignore some opportunities, and focus on specific parts
aim for specific improvements, then try to find ways to do archive it

6.8 SD (System Dynamic) model

defines product, release, iteration, backlog
defines error, change, approval rates at each step in the process
then simulates certain projects with different amount of unknowns
assumes unknown (unexpected) / known (pending) unknown types

pro agile
with high unknown rates agile outperforms
the higher the insecurity, the shorter release cycles should be

limitations agile
with no changes during the project non-agile is better
quality of result depends more on teamwork than on cycle times

6.9 agile culture

decision taking

frequent, fast, pluralistic, participative decision making
reassess decisions if new information arises
data-driven mindset

learning mindset
continuous self-improvement, feedback, learning from others
acknowledge the positive impact
possibly a reward system
dedicate time after each iteration to review

autonomy
shape own environment
participate in planning & choose own tasks
express disagreement towards business if required

supportive management
collaborate with subordinates
create appropriate environment for innovation & teamwork
encourage employees to be creative
set focus on satisfying customer requirements

team orientation
consult team members if issues appear
share responsibility of work with all members
praise performance on team level

personal communication
establish consensus with social interactions
work together closely on a daily basis
informal & face-to-face

sharing
share all information freely with others

ready for change
see certainty as impossible to archive
be comfortable with change, new ideas & new technology
accept & prepare for the iterative character of projects

6.10 teamwork

three fundamentals allow the big five of teamwork to happen
other contributing factors are team size/autonomy, meeting frequency, executive support, release cycles, know unknowns
lead to faster change analysis, decision taking & communicating changes

shared mental models
anticipate & predict other's needs
implicitly adjust strategies

mutual trust
share information
admit mistakes & accept feedback

closed-loop communication
acknowledge received messages
ensure sent messages were received
clarify that received message is same than intended one

team leadership
problem solving
set performance expectations & interaction patterns
synchronize efforts
enhance team functions
clarify team member roles
preparatory meetings & feedback sessions

mutual performance monitoring
identify lapses of other team members (over working, bad quality)
provide constructive feedback

backup behaviour
recognise if workload is badly distributed
shift work to underutilized members

adaptability
identify & assess changes, update project plan
improve repetitive practices

team orientation
weight different opinions & determine most correct one
be in an involving, sharing, participatory setting

7 terradine case studies

7.1 about company

creates testing equipment for chips
2004 market leader for systems on a chip SOC

semiconductor testing
test chips under extreme conditions for proper functionality
because manufacturing prone to errors
customers look for features, test speed, fast support

7.2 organisational issues

overworking, little training, but high employee loyalty
1990 total quality management TQM paradigm (all tasks tracked)
1996 revolutionizing product development RPD (better scheduling)
leading to drastically improved company as a whole

over commitment
300% utilization
introduced aggregate project planning APP
project have to fit in strategy, no over-committing anymore

badly organized projects
not well defined, scope creep introduced by engineers & marketing
milestones missed, sloppy project schedules, management could not react
no single person responsible as multiple divisions working on it
introduced phase-gate model

7.3 organisation structures

7.3.1 phase gate model

central plan all projects follow
clearly defined objectives for each team
after passing gate (decision to go on) the next phase is started

phase I
define market opportunity, customer requirements & feasibility
deliver market, technical, financial assessment, project plan
done with 2-6 people
gate checks deliverables, provides funding for phase II

phase II
refine product concept & create development plan
deliver specification, requirements, business, risk & development plan
done with cross functional core team & PM
gate & CEO reviews deliverables, provides funding, team commits

phase III
develop product to functional unit
deliver final unit, detailed documentation, analysis of customer market
done with full, cross function development team
gate reviews deliverables, then test parts & verify system

phase IV
verify product functionality & prepare shipment
deliver proof of manufacturing, final customer experience elements
done with full, cross function development team
gate reviews deliverables/plans, approves first customer shipment FCS

phase V
release product & processes to routine production, sales, support
deliver assessment, market analysis, future plans
done with normal, cross functional dev team
gate checks shipments & key issues, then frees up resources

7.4 jaguar project

customers want new product which can test multiple chips
stopped developing existing platform for this new universal one
but big risk to lose traditional customers, need to hit market window

7.4.1 project execution strategy matrix

define principles, processes & structure for project dimensions

definition dimension
broad project focus, but cost/time/benefit sweet spot (principle)
leverage existing technology, hierarchical objectives (processes)
core team & conner responsible (structure)

governance & staffing definition
staff works on project full time, core team responsible (principle)
hierarchical higher core team, APP to plan (processes)
organize in sub-teams, full time PM team (structure)

structure of activities
single development process, assigned to members (principle)
align to RPD framework (processes)
PM resource per team, reporting to jack (structure)

design, prototype, test
prefer simulation, else do physical (principle)
summarized monthly as risk (processes)
strategy owned by sub-project leader (structure)

senior management review/control
focus on metric gap analysis, on demand (principle)
monthly sponsor review, dash performance board (processes)
review with jack & mike bradley (structure)

real-time corrections
fixed at phase 2, monthly summary of trade offs (principle)
quarterly update of market, monthly update of key metrics (processes)
monthly core face-to-face, meetings with market team (structure)

7.4.2 organisation

division in sub project, each with own PM part of core team
meeting of core team weekly by telephone, monthly face-to-face

people
jack obrian (PM for three years, widely respected)
george conner (lead architect, prefers agile over used tools)
ed rogas (senior management, wanted people to use tools)

tools & processes
WBS (detailed description of task and relationship to others)
3-point expectation (minimum, expected, max time to complete task)
critical path analysis (uses WBS, 3-point estimation)
earned value analysis (measure progress by resource spending)
program managers for core team & sub-teams to keep team honest
analyzed CP weekly and tried to resolve issues
primavera had bad UX & was poorly understood by engineers

7.4.3 project performance

conflict at start because two division had their own plans
management provided as many resources as needed
fixed end-point to hit market window, to keep high tension

timeline
application specific integrated circuits (ASIC) on time
software development late because experienced staff not available
earned value charts clearly showed this behaviour, but PM in denial

alpha tech incident
big customer threatened to leave, put immense pressure on dev team
management committed to plan & monthly meetings with AlphaTech
real customer was motivating, but overworking, burn out
software team only fixed bugs
but the agreed on deadline was met

public release
software team was only fixing bugs, more staff thrown at problem
delayed for 6 months to stabilize software

7.4.4 reflections

project archived goals, but organisation was suboptimal
debatable whether tools were helpful or distracting
giebel found them helpful, but wrong usage
jack found them helpful, but criticised missing reactions to warnings
generally leaders found it helpful, but engineers thought its wasting time

primavera
bad UX, losing focus of big picture, some used other tools
lost personal drive for the success, collected wrong data