

Helsinki, September 23, 2002

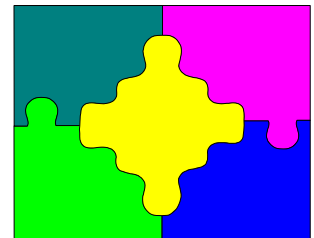
Opportunities and pitfalls in using extreme Programming

Karol Frühauf
INFOGEM AG, CH-5401 Baden, Switzerland
Secretary of the EOQ Software Group
Karol.Fruehauf@ACM.ORG

Copyright 2002 *INFOGEM AG*

Contents

- Opportunities of XP
- Traps of XP
- Pitfalls in using XP
- Conclusions



Copyright 2002 *INFOGEM AG*

Opportunities of XP



1. strong focus on quality
2. plan and design to cost
3. clear allocation of responsibilities
4. designed to cope with change
5. quick feedback on all levels

Copyright 2002 INFOGEM AG

Opportunity 1: Strong focus on Quality

- longing for pride by the craftsman acknowledged "internal quality" as measured by programmers
 - ⇒ not negotiable
- get rid of customer's fear to decide what he wants external quality as measured by customers
 - ⇒ negotiable



Copyright 2002 INFOGEM AG

Opportunity 2: Plan and design to cost

- resources are given, iteration (and release) schedule is defined, "internal quality" is not negotiable
- ⇒ scope is negotiated
- decompose stories into small tasks you can estimate, measure your productivity
- ⇒ get quickly faster and reliable in estimating

C. Barvotti's **PEOPLE**



"May I remind you that we have *deadlines* around here, Pembroke, and you should have been inked and shaded an hour ago!"

Copyright 2002 INFOGEM AG

Opportunity 3: Clear allocation of responsibilities

- customer has the duty to say what he wants and is responsible for the fitness for use of the product
- ⇒ defines scope as stories
- supplier has the duty to provide estimates and is responsible for the accuracy of the price
- ⇒ defines the (not negotiable) price

quality for the king
customer



E. Hürlimann

Copyright 2002 INFOGEM AG

Opportunity 4: Designed to cope with change

- it accepts requirements change as law of nature
 - customer has the right to change his mind, any time
 - ⇒ if he accepts the cost
- it justifies change of the solution if it fails to satisfy the needs
 - supplier has the right to update his estimates
 - ⇒ if he is prepared and able to learn



"I suppose we can take comfort in the fact the experts don't know what's going on, either."

Copyright 2002 INFOGEM AG

Opportunity 5: Quick feedback on all levels

requirements understanding
 questions regarding requirements
 correctness of the evolving
 solution

maintainability of the code
 correctness of the code
 programming progress
 iteration progress

technology questions

need for change

define test before you code
 on-site customer
 pair programming

pair programming
 automated unit testing
 automated unit testing
 daily stand-up meeting,
 continuous integration
 pair programming,
 daily stand-up meeting,
 XP room

on-site customer, XP room

Copyright 2002 INFOGEM AG

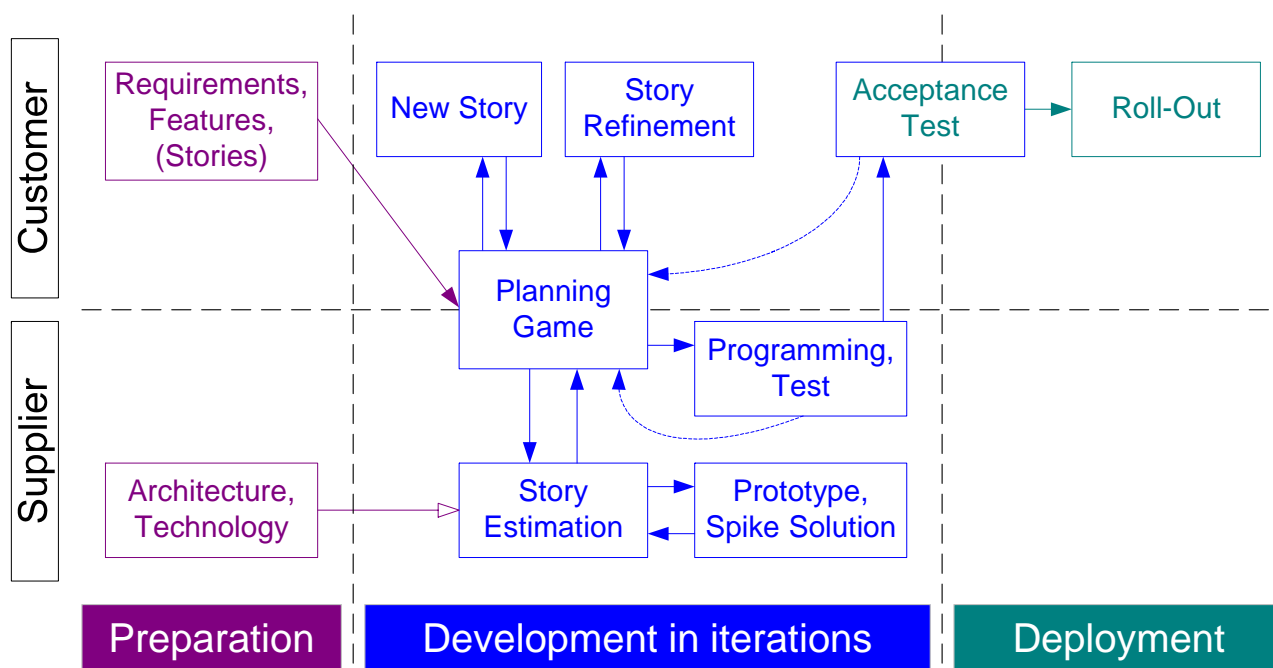
Traps of XP

1. XP covers only software development
2. XP covers software development completely
3. XP needs extremely good programmer



Copyright 2002 INFOGEM AG

Trap 1: XP covers only software development



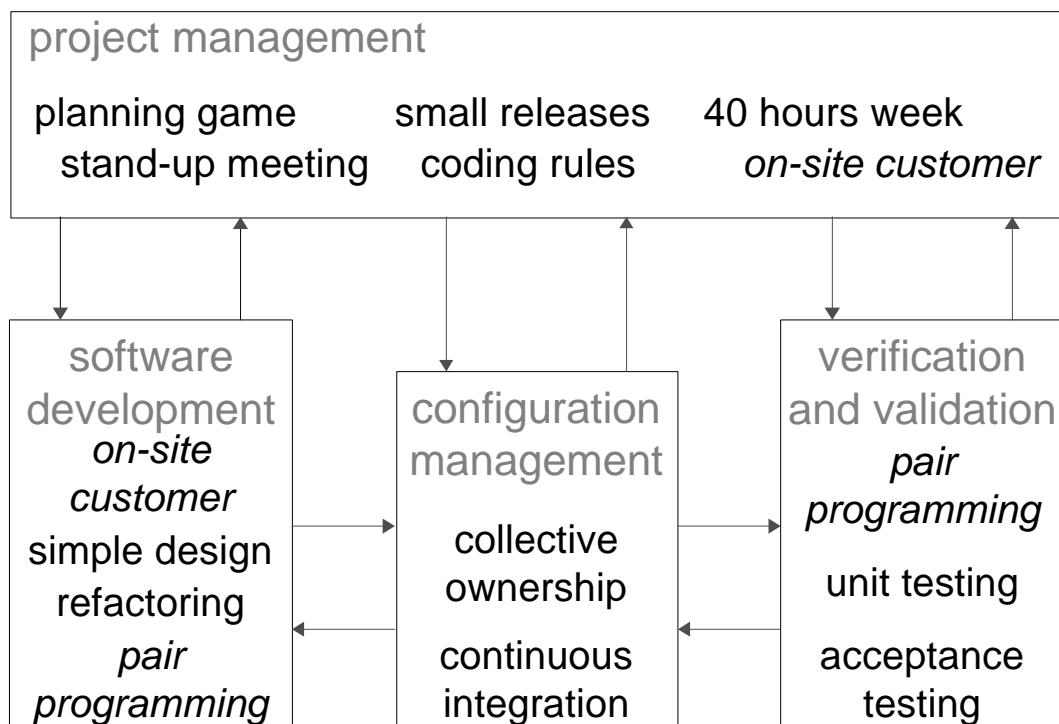
Copyright 2002 INFOGEM AG

Consequences of Trap 1

- How do you cope with business analysis? What story is it?
 - How does the customer budget the project?
 - How does the supplier estimate the cost?
 - How does a contract looks like?
 - How does the software get distributed, installed, started up and operated?
 - How does problem management work once the system is up and used?
 - Is customer or acceptance tester or planner or a programmer addressed by help desk?
 - Are problems stories?
 - Is there something like a patch?
- ⇒ it looks similar to maintenance but is not

Copyright 2002 INFOGEM AG

Trap 2: XP covers software development completely



Copyright 2002 INFOGEM AG

Consequences of Trap 2

How are the dependencies to other systems made visible? Are (dying) stories the right thing for that?

How is the product integrated with other products?

How do you recover requirements from test cases and programs?

How do you define test completeness without updated stories?

Why bother tracing test cases to stories if they get destroyed?

How do you track overall progress with no estimates for the not planned and not invented yet stories?

Trap 3: XP requires extremely good programmer

"you need good people"

- for good people any method is suitable or good people make choose always the adequate method
- where do you get up to ten good people?
- ⇒ good prerequisite is to have persons in team
- + with the same or very similar educational background
- + with roughly the same communicational skills
- + possessing identical degree of egoless behaviour
- + with similar attitude concerning discipline

Pitfalls – a discussion

the following slides sketch a case study

- I would like to discuss with you the suitability of the XP approach for this case
- ⇒ Where can you justifiably expect pitfalls?
- ⇒ Where do you think XP will work?



Copyright 2002 INFOGEM AG

Case study (1)

in-house project 9999++

goal

replace existing individual solution by a standard ERP solution
because the maintenance contract can't be prolonged any more

time constraints

- module HR operational 1.1.2003
- modules FI and CO operational 1.7.2003 (start business year)

scope constraints

- the current functionality of the system must be available
- the requirements resulting from the Business Reengineering project in the department PA have to be implemented
- all other processes have to be implemented more robustly and efficiently
- the system must be adapted to accommodate the change of the external interface to the governmental body XY

Copyright 2002 INFOGEM AG

Case study (2)

technological constraints

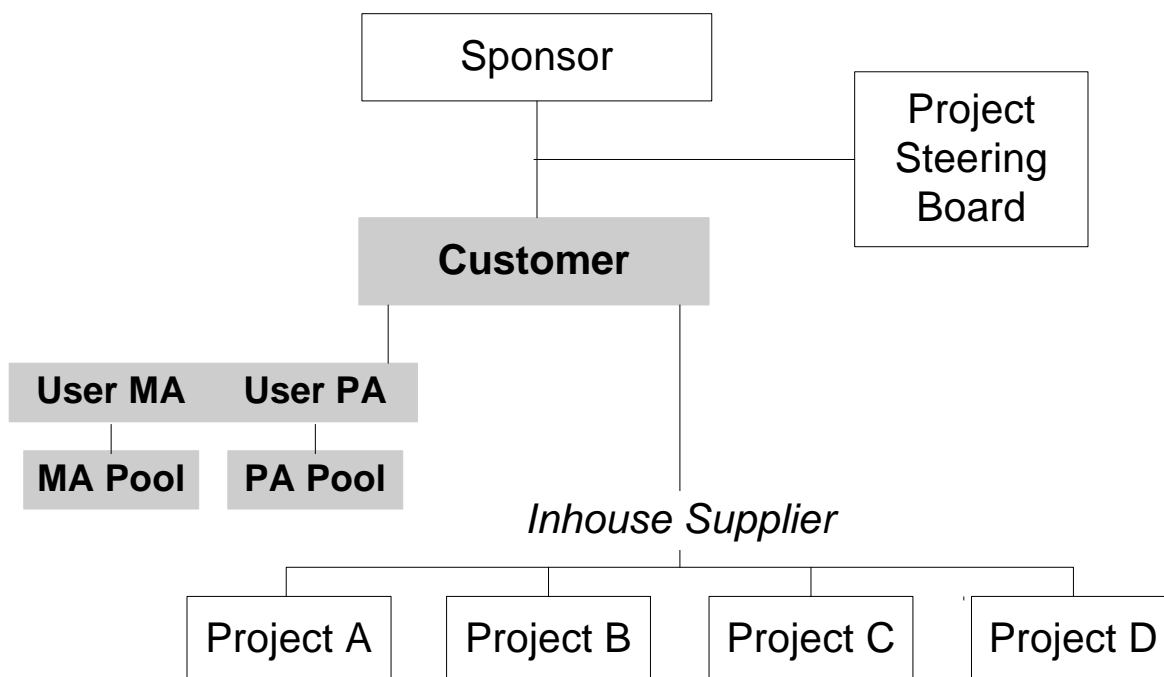
- programming language: new
- operating system: new
- data base management system: new
- report generator: new

organisational constraints

- sponsor = member of the executive board
- on-site customer = no user can play the role of the
- "customer project" 9999 plays the role
- project leader 9999 = lead of the IT department
- additional responsibility: co-ordinate the projects
- project steering board started 1.7.2002
- a number of other projects need to be synchronised but is not involved
- basic services group delivers but is not integrated in the project

Copyright 2002 INFOGEM AG

Case study (3)



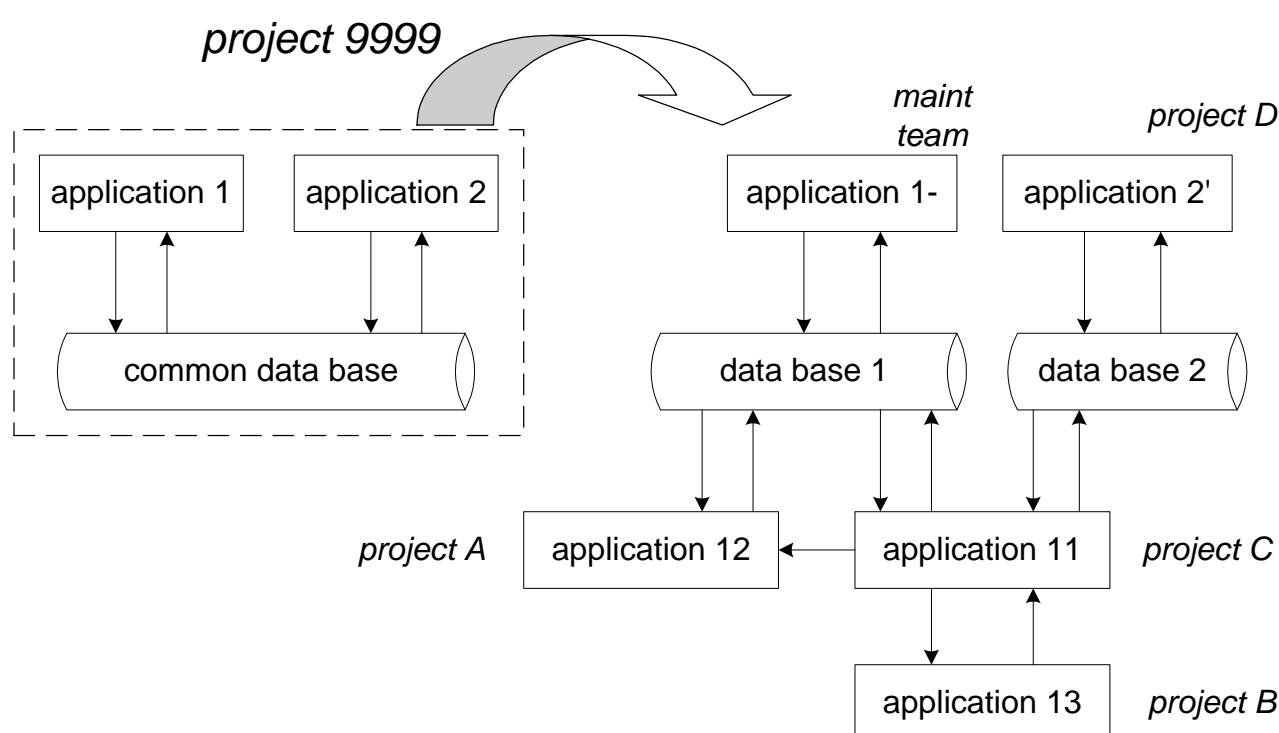
Copyright 2002 INFOGEM AG

Case study (4)

- the projects started 1.7.2001 – they are projects in their own right
- the project team size varies between 1 and 5 persons
- an architecture project started 1.9.2000 and completed 1.9.2001
the technological foundation for the future solution
- no application architecture is defined
- the project 9999 started 1.4.2002
- every project has a defined scope and specifies its external interfaces
- every project has two customers: project 9999 (which stories) and a user representative (what is in the story)
- the group of projects 9999+ is technology driven – users don't suffer enough with the current solution in order to take initiative

Copyright 2002 INFOGEM AG

Case study (5)



Copyright 2002 INFOGEM AG

Case study (6)

methodological constraints

- in the software development use of XP is mandatory
- this is the first set of projects using XP
- not all XP practices are introduced yet
 - iteration planning games are used extensively
 - no stand-up meetings
 - pair programming rarely applied
 - automated unit testing
 - continuous integration with nightly build without built in unit tests
 - iterations are not usable in production (yet)
- no acceptance tester, no acceptance test of iterations
- project customising the standard ERP product follows own method
- general project planning and controlling mechanisms have to be used

Copyright 2002 INFOGEM AG

Conclusions

Did our
software
development
method apple
fall down already?
Do we still ponder?



Theorist Newton pondering the apple's fall
Blissfully unaware of all the controversy.

Or did somebody
even discover
already the
gravitation law
of the
software
development
methodology?

Do you know him?
Do you know the
law?

Copyright 2002 INFOGEM AG

Effects of process anorexia

- + shorter time to usable results
- + loss of complexity (by gaining simplicity)
- + shorter (and less expensive) defect removal loops
- + less work products in need of configuration and change
- + gain in productivity
- + gain in continuous product improvement
- o amalgamation of maintenance and development
- shorter product perspective
- loss of independence from product information storage resources
- loss of product correctness visibility
- loss of stability in the user environment

Conclusions

- o there is neither a heavy weight / robust nor a light weight / agile silver bullet
- o the method needs to satisfy the requirements on the process (results)
- o it is easier to fit the methodology to people than the other way around
- o the vision and spirit of the methodology is what counts, not the details
- o many successful projects employed a coach in order to avoid ...