

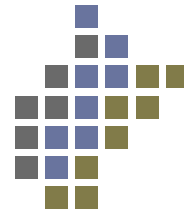


**ABB**

1st  
International  
Global Requirements Engineering  
Workshop

**GREW'07**

Workshop Wrap-up



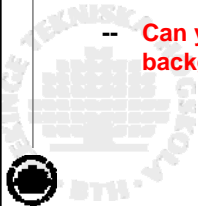
**Tony Gorschek**  
Blekinge Institute of Technology  
Software Engineering Research Lab (SERL)

1st  
International  
Global Requirements Engineering  
Workshop



**Session 1 Product Lines**

- **Problems of scale (tools and techniques from academia disintegrate when applied in industry e.g. 1000000 requirements)**
- **Important to store decision rationale (how/what/lvl of formalism?)**
- **Tools – Start with people and processes then tools, but tools can be a good way to initiate change (raise awareness of needs)**
  - **Since RE (Product Line eng.) spans e.g. marketing, engineering etc important to make sure that tools and data stored is relevant for different roles**
- **Can you "toolify" domain knowledge to enable knowledge and background sharing/homogenization**



## Session 2 Global Communication

- Weak processes in a company (relying on informal communication and domain knowledge) will lead to failure when distributing the development **but** the intro. of distributed development may stimulate awareness (and improve?) processes
- Cultural differences (increases demands on competence of leadership, e.g. project managers to catch problems early)
- Instant feedback very important (iterative?!) + assuring that delays aren't showstoppers
- Incomplete specifications (assumptions of domain knowledge, background, education etc aggravates the potential for failure?!)
- Increased efforts in some key areas like RE and PM, and different **EFFORT DISTRIBUTION** can be a result (or a necessity?!) of distributed development
- Lean and Rich media (combination is best... effective but maybe would be interesting in the light of efficiency)
- Developer experience and communication structure may help in mitigating problems of awareness in distributed projects



## GENERAL

- Is global development unique enough to be differentiated from general development?
- Taxonomy (vocabulary?!) that structures research attempts in GSE/GRE? Define GSE/GRE?!

## Ending soon...

- Plan:
  - GREW'08
  - Wicki
  - homepage: [www.bth.se/grew07](http://www.bth.se/grew07) or later [www.bth.se/grew](http://www.bth.se/grew)
- Feedback, please send ideas and feedback to [tgo@bth.se](mailto:tgo@bth.se)



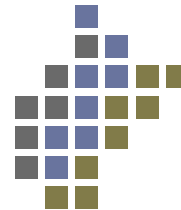


**ABB**

1st  
International  
Global Requirements Engineering  
Workshop

**GREW'07**

Workshop Wrap-up



**Tony Gorschek**  
Blekinge Institute of Technology  
Software Engineering Research Lab (SERL)

1st  
International  
Global Requirements Engineering  
Workshop



**Session 1 Product Lines**

- Communication
- Organization
- Synchronization
- CM (have to timeliness, variants, sets etc)
- Stakeholder specific views (workable sets that is end-user friendly for day-to-day work)
- Req. management
  - **Problems of scale (tools and techniques from academia disintegrate when applied in industry e.g. 100000 requirements)**
  - Multiple regions don't want to be under the control of central authority e.g. add new req. instead... Holistic view when developing tools/techniques and PROCESSES



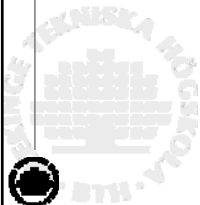
## Session 1 Product Lines

- What do you do with all the captured information? E.g. formally specify it...?
- Important to store decision rationale – tools don't store rationale in formal specification e.g. rich tracing not implemented in tools today...
  - How do you remember and then find an discussion and a decision (rationale)
  - Catch / Store / Track the important ones?
  - Reuse of rational, but also best-before date
  - How the information Goes in – and – comes out... (two separate problems....)? E.g. making sure that there are processes are in place to store the information, and have good possibilities to search for information
  - People are the starting point, start with them then processes
- What is unique for product lines and variability management?
- Process for PL has to span marketing and management (use different tools and views problems in different ways), tools also have to span the gap...



## Session 1 Product Lines

- Tools, political decisions to buy? Tool that can hold all views, but be simple?
  - Filters to present views, but what if the filter is wrong? Input and data separation – will it be input correctly to support views?
  - Maybe everyone should see not only one view but multiple views (share understanding and share knowledge)...
- Can you "toolify" domain knowledge to enable knowledge and background sharing/homogenization



## Session 2 Distributed Communication

- Communicate the right information to the right person in time – need for a good doc management system and also good filtering of information avoiding spam
- Weak processes in a company (relying on informal communication and domain knowledge) will lead to failure when distributing the development
- Lack of Situational awareness = damages trust btw dev and customer
- Cultural differences → Need to know how issues will be communicated and resolved (how to catch a potential problem early)
- Skill of Project management is vital in distributed development – especially when it comes to catching potential problems (collocated projects can compensate for bad management – but does not work in distributed teams)
- Instant feedback is very important in distributed teams, how and what mechanisms are there!?
- Increased efforts in some key areas like RE and PM, and different EFFORT DISTRIBUTION
- Incomplete specifications (assumptions and background tacit knowledge can not be assumed)
- Staff turn-over increases with distributed development



## Session 2 Part II Distributed Communication

- Realistic in terms of the assumptions e.g. that models exist for all things and that they are relevant and up to date
- Models need to be closer to reality, e.g. what is used – in addition
- Model hard and easy stuff, i.e. no meaning to model the easy stuff – thus you can't assume e.g. automatic consistency checks etc... in reality you model only hot-spots...
- Is there any pros. to formalize/detailed document items in global dev? → How much effort and what will it cost me if I don't?
- Important to document decision rationale!
- Distributed environments can motivate the improvement of processes and improved documentation...
- Multidimensional views (e.g. not only project but product, or product portfolio, company etc...)
- Reward e.g. developers for successful projects/products... incentive driven to get better at e.g. documentation because they see the point...
- What type of "motivation" is relevant? E.g. money, but maybe allowing a dev. to do a good job... or do interesting work...
- How can we bring the customer closer to the developer, e.g. let the developers approach the customers?
- Scenario based and enriched media can be used to communicate btw cust. and dev.



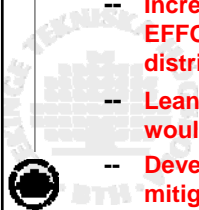
### Session 1 Product Lines

- Problems of scale (tools and techniques from academia disintegrate when applied in industry e.g. 1000000 requirements)
- Important to store decision rationale (how/what/lvl of formalism?)
- Tools – Start with people and processes then tools, but tools can be a good way to initiate change (raise awareness of needs)
  - Since RE (Product Line eng.) spans e.g. marketing, engineering etc important to make sure that tools and data stored is relevant for different roles
- Can you "toolify" domain knowledge to enable knowledge and background sharing/homogenization



### Session 2 Global Communication

- Weak processes in a company (relying on informal communication and domain knowledge) will lead to failure when distributing the development **but** the intro. of distributed development may stimulate awareness (and improve?) processes
- Cultural differences (increases demands on competence of leadership, e.g. project managers to catch problems early)
- Instant feedback very important (iterative?!) + assuring that delays aren't showstoppers
- Incomplete specifications (assumptions of domain knowledge, background, education etc aggravates the potential for failure?!)
- Increased efforts in some key areas like RE and PM, and different **EFFORT DISTRIBUTION** can be a result (or a necessity?!) of distributed development
- Lean and Rich media (combination is best... effective but maybe would be interesting in the light of efficiency)
- Developer experience and communication structure may help in mitigating problems of awareness in distributed projects



## Session 2 Global Communication

- Weak processes in a company (relying on informal communication and domain knowledge) will lead to failure when distributing the development but the intro. of distributed development may stimulate awareness (and improve?) processes
- Cultural differences (increases demands on competence of leadership, e.g. project managers to catch problems early)
- Instant feedback very important (iterative?!) + assuring that delays aren't showstoppers
- Incomplete specifications (assumptions of domain knowledge, background, education etc aggravates the potential for failure?!)
- Increased efforts in some key areas like RE and PM, and different EFFORT DISTRIBUTION can be a result (or a necessity?!) of distributed development
- Lean and Rich media (combination is best... effective but maybe would be interesting in the light of efficiency)
- Developer experience and communication structure may help in mitigating problems of awareness in distributed projects



## • Session 1 Product Lines

- P2: Using a meta model (modeling) to handle variability in global RE
  - Rationale has to be multidimensional depending on e.g. country/market
  - One set of rationale per market, and then instantiate the product, core assets?
    - Communication
    - Organization
    - Synchronization
    - CM (have to timeliness, variants, sets etc)
    - Stakeholder specific views (workable sets that is end-user friendly for day-to-day work)



## Session 2 Distributed Communication

-- P4 Looking at a novel communication mode to see if it affected req. negotiations (mix of rich and lean media and the impact of this...) Experiment performed using student developers to compare – RESULT: A mix is best acc. to the experiment

QA:- Multi or single threaded? = Single threaded (discussion forum)

- (Time? Efficiency?) (Groups: Rich vs. Lean and study time too)
- Effectiveness studied, but effectiveness needs to be studied too...
- Not only for RE or even distributed dev, but also for any dev. activity and for collocated teams (video -> face to face meetings)
- One-to-one or group to group?
- Dependencies btw issues? Yes, there where...

-- P5 Awareness study in two globally distr. industry projects (factors: distance, awareness, communication structure)

- LL:
- Experienced team members bridge the awareness gap
  - Centralized structure keep new teams aware
  - Frequent meetings improve awareness among local and remote teams
  - Multiple tasks given to dev. reduces the effect of delays from coordination

- Disc:
- all of the issues can be seen in collocated project too...
  - tools differed btw teams...



## Session 2 Part II Distributed Communication

-- P3 Empirical study of distributed setting  $\leftrightarrow$  RE practice using 3 dimensions (physical, temporal, organizational)

- Physical and temporal distribution increases the awareness of organizational distribution
- Social distance (Culture, Lang, Process, View)
- Considering Interface “Strings”
- Describe interdependencies btw RE Practices

-- P6 RP in Distributed Projects

- extend sysifus with knowledge nuggets (cso \* sys mod element) to model multiple releases
- Traceability btw release items and model
- Support distr. dev. through e.g. making sure that if a release item disappears, so does the model

