



The Role of Configuration Management in Outsourcing and Distributed Development

Lars Bendix
bendix@cs.lth.se

Christian Pendleton
Christian.Pendleton@softhouse.se

*Department of Computer Science
Lund University
Sweden*

*Softhouse Consulting
Malmö
Sweden*



Distribution



Distribution is good:

- gives a larger pool of talents and specialists
- allows for cooperation between departments/companies
- facilitates integration for mergers and acquisitions
- allows for around the clock work
- gives more flexibility in scaling up and down projects

Distribution is bad:

- it is more complex to manage
- it creates silos between groups
- people don't understand and trust each other
- you lose control over remote teams/people
- ...

Does it really have to be that way?



CM and Distribution



Configuration Management already handles “distribution”:

- programmers are rarely co-located
- developers are often distributed (also in time)
- we handle development AND maintenance
- where is the Project Manager?



Our goals



What are we trying to obtain:

- make sense of DD:
 - is there something here we don't understand?
 - is there something that others have overlooked?
- what special challenges are there in DD:
 - which ones involve CM?
 - which ones do not involve CM?
- how can challenges be alleviated by CM support
 - “same old stuff”
 - re-think implementation
 - oops – that's a new one ;-)



The method



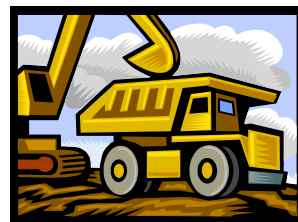
Challenges



CM in DD



Cases



Analysis



Discussions





Prior art



ICGSE >100 papers
... and beyond (Google Scholar)

Challenges:

- da Silva et al., 2010 (54 papers)
- Jiménez et al., 2009 (78 papers)

CM & DD:

- Pilatti et al., 2006 (4 cases)
- Fauzi et al., 2010 (24 papers)



Challenges 1



da Silva et al., 2010:

- effective communication
- cultural differences
- coordination
- time zone differences
- trust
- asymmetry in processes, policies and standards
- physical distance
- IT infrastructure
- different knowledge levels or knowledge transfer
- tracking and control
- cooperation
- people management/conflict resolution
- language barriers
- task allocation
- identification of roles and responsibilities
- knowledge management
- scope and change management
- overall visibility
- differences in technologies used
- creating team spirit
- project planning
- quality
- intellectual property issues/ confidentiality and privacy
- different stakeholders
- schedule management
- synchronizing work between distributed sites



Challenges 1



da Silva et al., 2010:

- ***effective communication***
 - cultural differences
 - ***coordination***
 - *time zone differences*
 - *trust*

 - asymmetry in processes, policies and standards
 - *physical distance*
 - IT infrastructure
 - *different knowledge levels or knowledge transfer*
 - ***tracking and control***
 - ***cooperation***
 - people management/conflict resolution
- language barriers
 - *task allocation*
 - identification of roles and responsibilities
 - *knowledge management*
 - ***scope and change management***
 - *overall visibility*
 - *differences in technologies used*
 - creating team spirit
 - *project planning*
 - quality
 - *intellectual property issues/ confidentiality and privacy*
 - different stakeholders
 - *schedule management*
 - ***synchronizing work between distributed sites***



Challenges 1



da Silva et al., 2010:

- **effective communication**
- cultural differences
- **coordination**
- **time zone differences**
- **trust**
- asymmetry in processes, policies and standards
- **physical distance**
- IT infrastructure
- different knowledge levels or knowledge transfer
- **tracking and control**
- **cooperation**
- people management/conflict resolution
- language barriers
- **task allocation**
- identification of roles and responsibilities
- **knowledge management**
- **scope and change management**
- overall visibility
- **differences in technologies used**
- creating team spirit
- project planning
- quality
- intellectual property issues/ confidentiality and privacy
- different stakeholders
- schedule management
- **synchronizing work between distributed sites**



Challenges 2



Jiménez et al., 2009:

- ***communication***
- *group awareness*
- ***software configuration management***
- *knowledge management*
- ***coordination***
- ***collaboration***
- *project and process management*
- process support
- *quality and measurement*
- risk management



CM & DD 1



Pilatti et al., 2006:

- *minimize dependencies between distributed teams*
- ***work with one instance of SCM environment***
- ~~all CIs required for a build should be put under CM~~
- ~~projects should define one build coordinator~~
- ~~establish and clarify CM before starting project~~
- ~~CM engagement in the beginning should be prioritized~~
- ~~always plan and document baselines (in CM plan)~~
- *re-plan activities due to scope floating across teams*



CM & DD 2



Fauzi et al., 2010:

- ***dispersed software teams do not get information on what other teams are doing***
- ***difficult to know the traceability of each module***
- *the definition of modifications or problems to be handled is unclear*
- *dependency*
- ~~delay and increased time required to complete change requests~~
- ***working in different SCM environments***
- ~~change requests are handled at various levels in the project~~
- ~~lack of a planned baseline~~
- lack of coding standards
- *code ownership*
- unclear flow of development
- *tool selection*
- ***artefacts with different versions and content at each site***



Categorizations



Configuration Management involvement in DD challenges:

- not related
- *weakly related*
- ~~***strongly related, but not particular to DD***~~
- ***strongly related***



Normalization



But something didn't quite work:

- there were too many challenges
- granularity was too varied
- some were broad-sweeping and imprecise

So we iterated over the original challenges:

- unify identical or similar challenges
- rename (and redefine) challenges to match CM terminology
- remove “irrelevant” challenges
- aggregate smaller challenges into larger, coherent chunks
- 11-17 resulting normalized challenges



Results 1



Not related to Configuration Management:

- code ownership
- need of office space
- application of an iterative agile process
- different governments, laws, rules and regulations
- unclear flow of development
- lack of coding standards
- different stakeholders
- quality
- creating team spirit
- identification of roles and responsibilities
- people management/conflict resolution
- IT infrastructure
- process support

• risk management



Results 2



~~*Co-located Configuration Management challenges:*~~

- lack of baselines
- all CIs required for a build should be put under CM
- establish and clarify CM before starting project
- CM engagement in the beginning should be prioritized
- difficult to know the priority of each module



Results 3



Weakly related to Configuration Management:

- project management
- trust
- intellectual property issues



Results 4

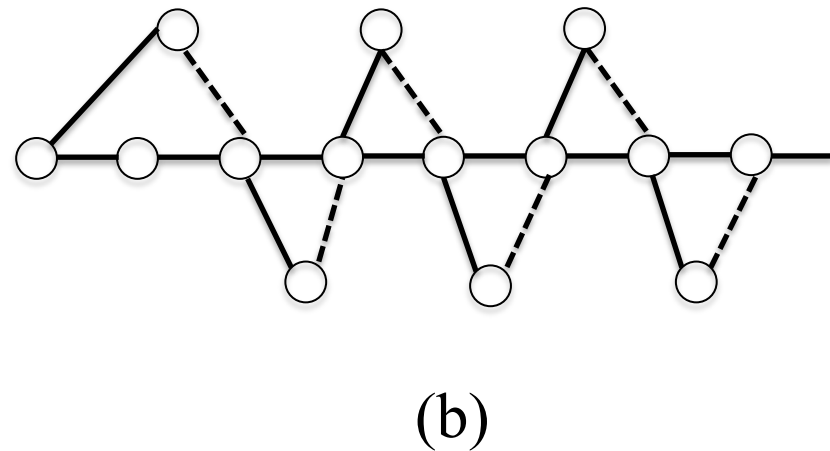
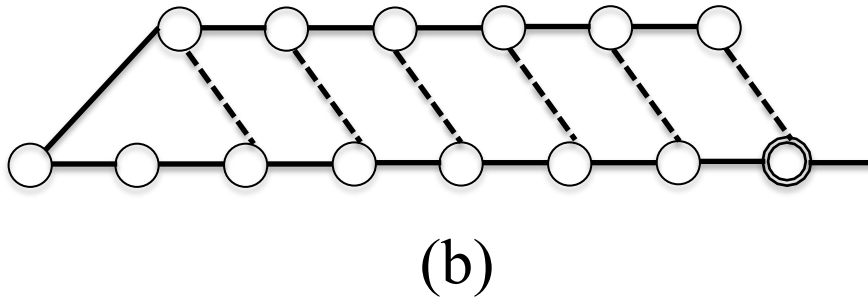
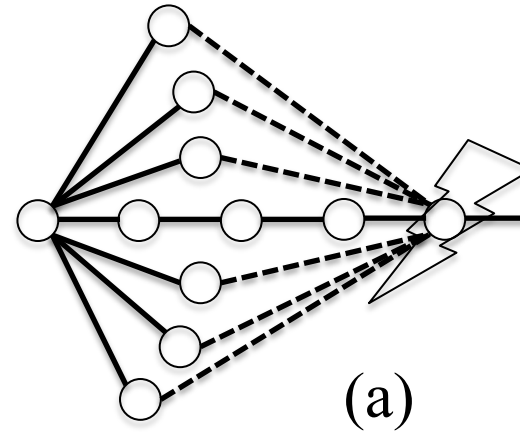
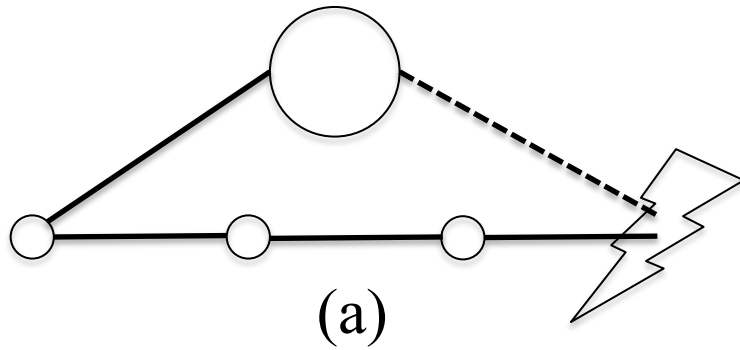


Strongly related to Configuration Management:

- communication
- co-ordination
- change management
- collaboration
- one SCM environment
- knowledge management
- (virtual) team awareness



Co-ordination (*strongly related*)

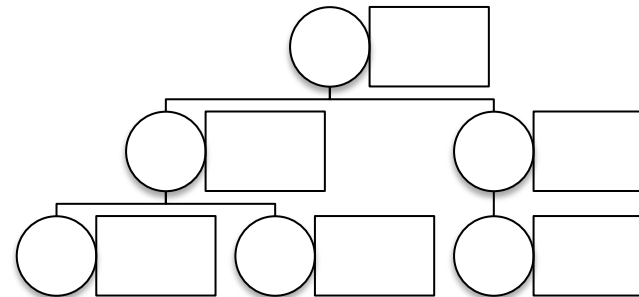
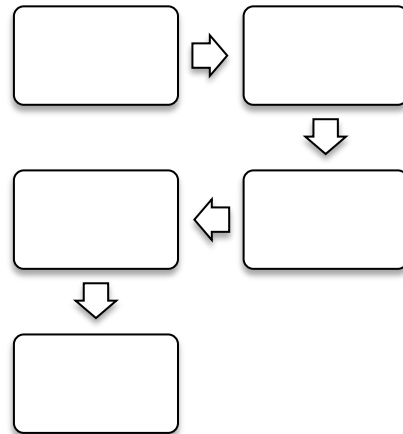




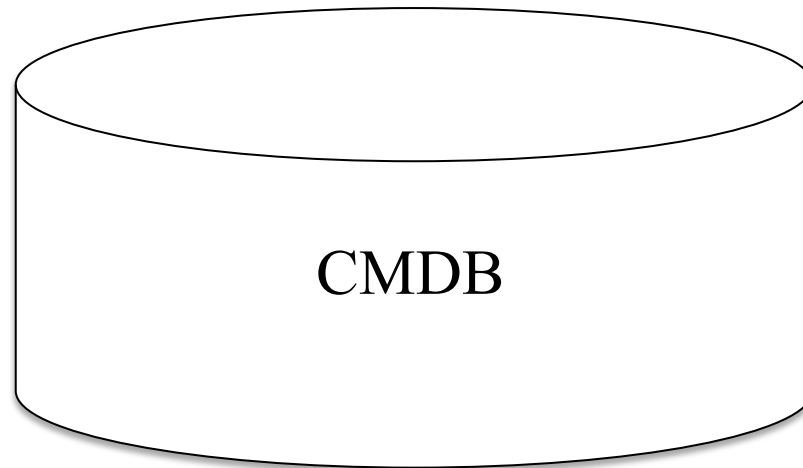
Project management (*weakly related*)



handover
automation



dependencies
relations
architecture

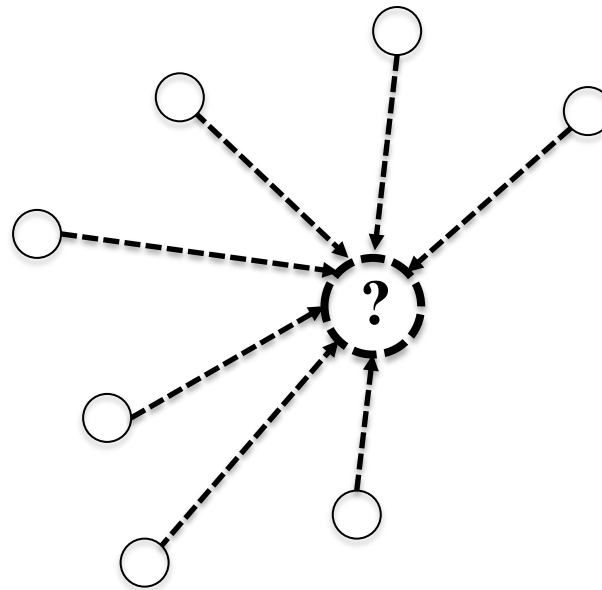




Team awareness (*strongly related*)



Hypothetical
Continuous
Integration





Our mission



We were asking ourselves:

- was there something here we didn't understand? YES!
- was there something that others had overlooked? YES!

Does it really have to be that way? NO!

Who can benefit:

- project manager
- programmer
- developer
- requirements engineer
- project/product owner
- configuration manager



Take-away I



Configuration Management involvement in DD challenges:

- *not related*
- **weakly related**
- *strongly related, but **not** particular to DD*
- **strongly related**



Take-away II



Dear CM, here are our lists of:

- distributed challenges you don't want to hear of
- distributed challenges someone ought to be ashamed of
- distributed challenges where you can add some value
- distributed challenges that you are expected to fix (where you will have to re-think implementation)
- distributed challenges that you did not know you could fix (however, you will need some help – future work)

http://fileadmin.cs.lth.se/cs/Personal/Lars_Bendix/Research/SCM4GSD/