

He (Jason) Zhang, Ross Jeffery, and Liming Zhu

Hybrid Modeling of Test-and-Fix Processes in Incremental Development

- *International Conference on Software Process 2008*



Australian Government
Department of Communications,
Information Technology and the Arts
Australian Research Council

NICTA Members



The University of Sydney



Queensland
Government



Griffith
UNIVERSITY



QUT
Queensland University of Technology



THE UNIVERSITY
OF QUEENSLAND
AUSTRALIA

NICTA Partners

- Introduction
- Hybrid process simulation modelling
- Model background
- Modelling incremental development
- Case example
- Conclusion



- Software process and process simulation
 - Osterweil identified two complementary types of software process research, i.e. *macro-process* level and *micro-process* level.
 - The *continuous* and *discrete* process modeling and simulation approaches relate to these two types of research.
- Hybrid process simulation modeling (HPSM)
 - HPSM is possible to avoid the limitations of applying single modeling method, and more realistically capture complex real-world software processes.
 - The systematic review identified that system dynamics (SD) plus discrete-event simulation (DES) has been the most common combination for constructing hybrid software process simulation models.

- Two approaches to construct a hybrid process model
 - **Vertical Integration:** primarily implements *discrete* modeling at the lower (micro-) process level, then *continuously* calculates the process factors and incorporates the feedback loops at system (macro-process) level;
 - **Horizontal Integration:** the sub-processes within a large scale software process may be modeled using different approaches *respectively* and *sequentially*, these sub-processes are connected by predefined phase transitions.
 - Most of previous work in HPSM concentrated on vertical integration.

- Martin and Raffo's model
 - Developed to maintain consistency between the discrete activities and the change of continuous portions.
 - Allows investigation of the effects of discrete resource changes on continuously varying productivity, and the influences of discrete tasks.
- Rus' and Lakey's models
 - Modeled as a set of discrete activities..
 - Such approximation of continuous models is coarse, and the time advance might be different between discrete phases.

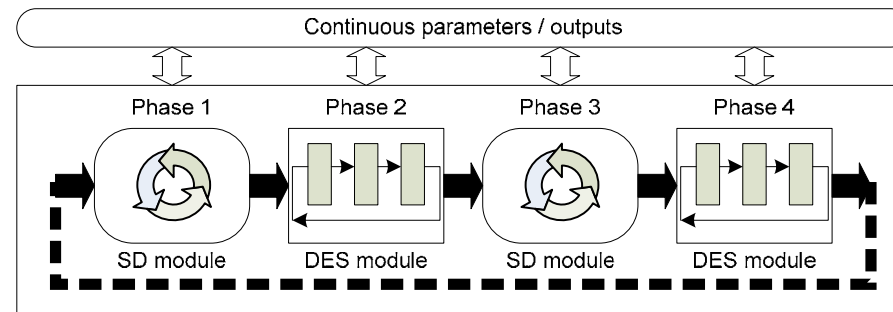
- Choi's model

- It enables modeling differential equation system with discrete-event by constraining the time advance of discrete-event model to be a small enough constant-time. Enhancement of Rus & Lakey.

	Martin & Raffo	Rus <i>et al.</i> & Lakey	Choi <i>et al.</i>	Zhang <i>et al.</i>
Model framework	Continuous	Discrete-event	Discrete-event	Discrete-event
Module/phase level	Discrete-event	Continuous	Continuous	Continuous/Discrete-event
Sim tool	Extend	Extend	DEVSim++	Extend

- Limitations of existing vertical integration
 - Emphasize and apply *continuous* and *discrete* modeling at different levels.
 - Details in every single phase (sub-process) of the modeled process are necessary for modeling, no matter where the investigation focuses.
- Questions for choosing modeling approach at phase level:
 - What is the purpose of modeling and simulation of this module/phase?
 - Which portion of the investigated process(es) is the focus of our study?
 - What are the observed factors in this portion related to model outputs?
 - At what granularity level we possess the information of this phase?

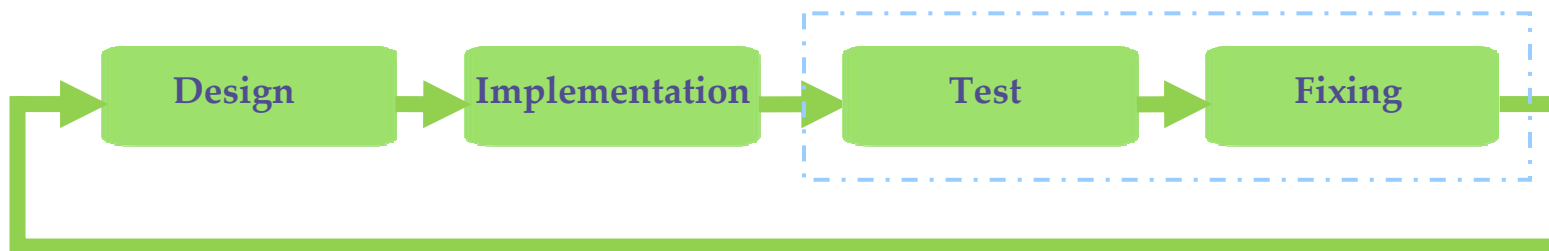
- A flexible hybrid modeling scheme using horizontal integration



- Module level: flexible selecting *continuous* or *discrete* modeling approach.
- Model level: connecting *discrete* modules and *continuously* observing important parameters.
- Maintain model integrity by carrying out fine-grained investigation on portion of process while allowing the limited knowledge on others.
- Enable the effective model reuse by assembling modules for process simulation.

- Incremental development

- It divides the development into several smaller increments, which are gradually accumulated to become the complete system.

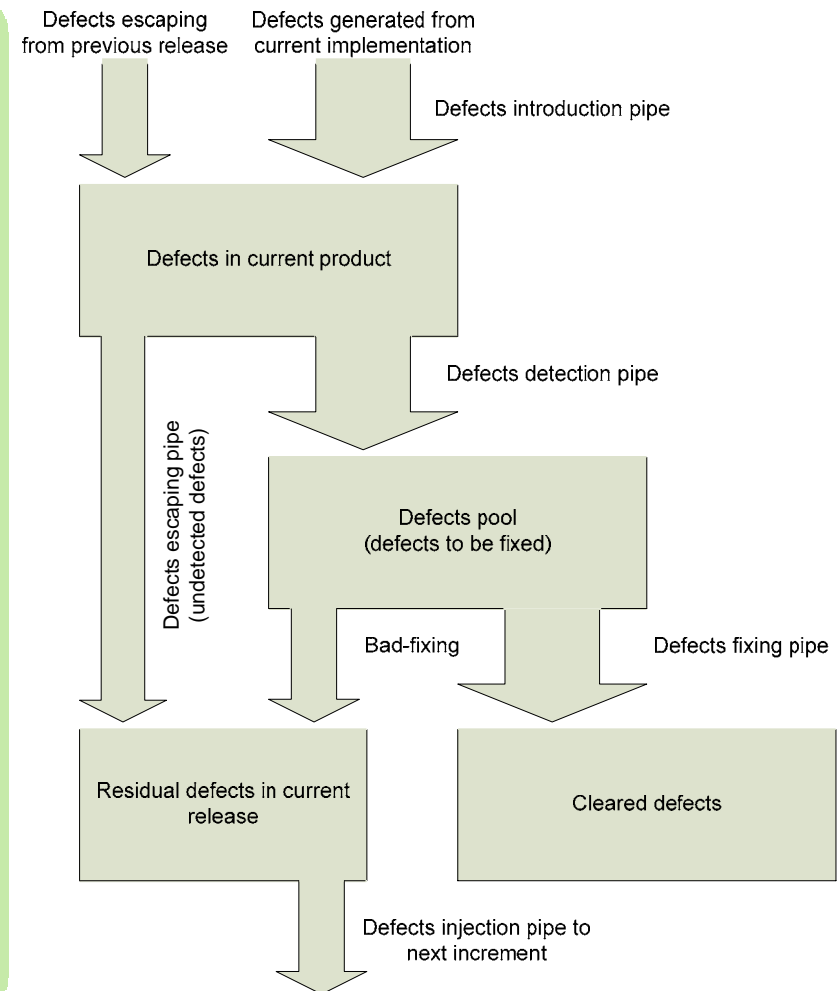


- Process characteristics:
 - transition, iteration, and concurrency;
 - every single phase in each increment cannot be omitted in simulation.
- Investigation focus: test-and-fix process in incremental development with quality and productivity concerns.

Background – quality conceptual model



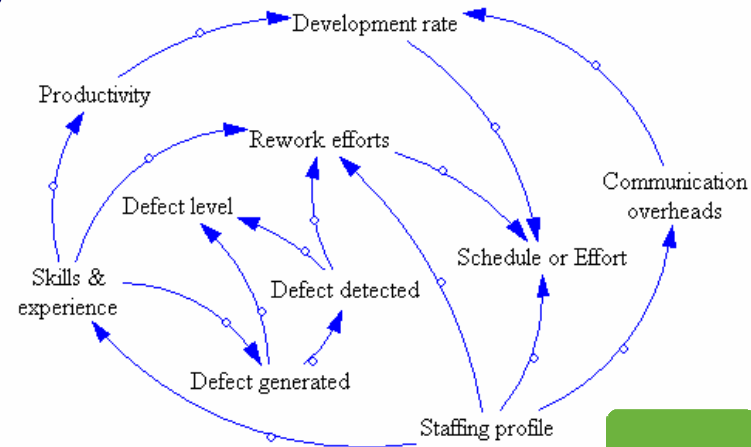
- “*Tank and pipe*” model
 - We assume that any reduction in software defects remaining in a product improves its quality.
 - We developed the conceptual model based on Boehm and Jones quality models, but focusing on test-and-fix process of incremental development.
 - Defects flow into a cascade of tanks through various defect source and transferring pipes.
 - Residual defects of current release will be input to next increment.



- Abdel-Hamid and Madnick's model
 - It is based on the waterfall process model; did not isolate the newly generated defects and the residual defects from the test-and-fix process.
- Huff's model
 - It did not identify the reproduction of residual active defects in the succeeding increments.
- Tvedt's model
 - It did not explicitly handle the different types of defect between increments.
- Cangussu's model
 - It is based on control theory; did not identify defect categories; ignored the influence between implementation and test-and-fix processes.

- Approach for module: system dynamics

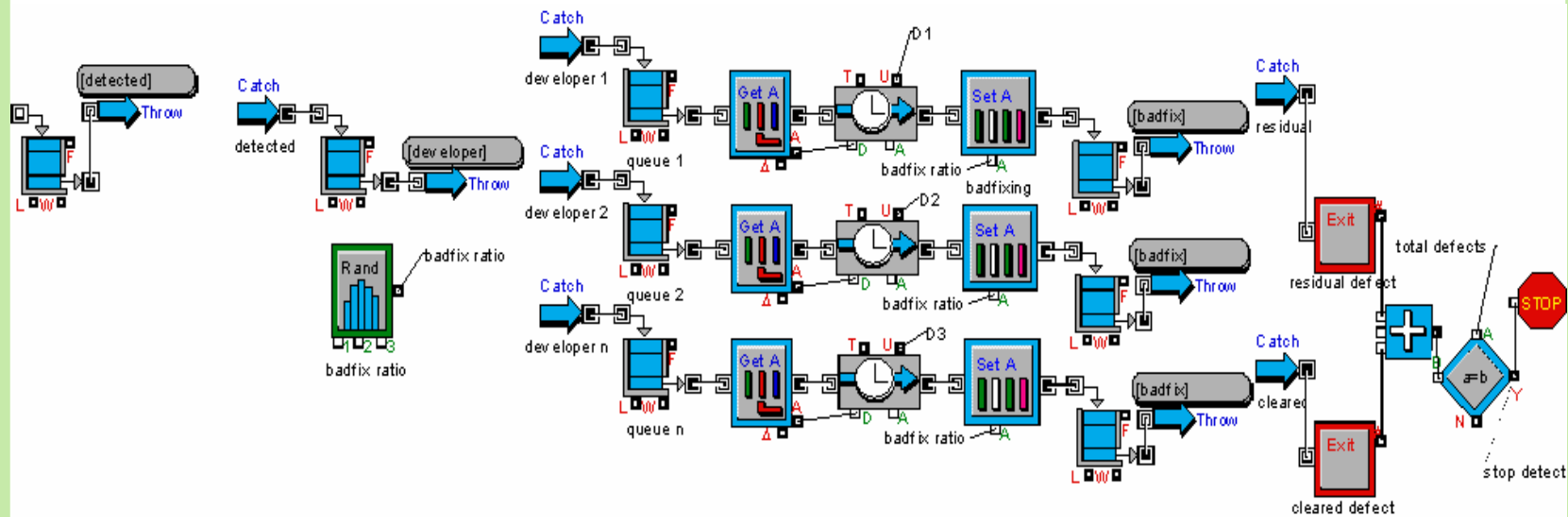
continuous feedback structure of generic development process



- Basic assumptions:
 - System is implemented with an incremental top-down strategy;
 - Most defects in early releases committed in the core components and become active.
- Defect sources during implementation:
 - new defects + residual defects + reproduced defects by active ones.

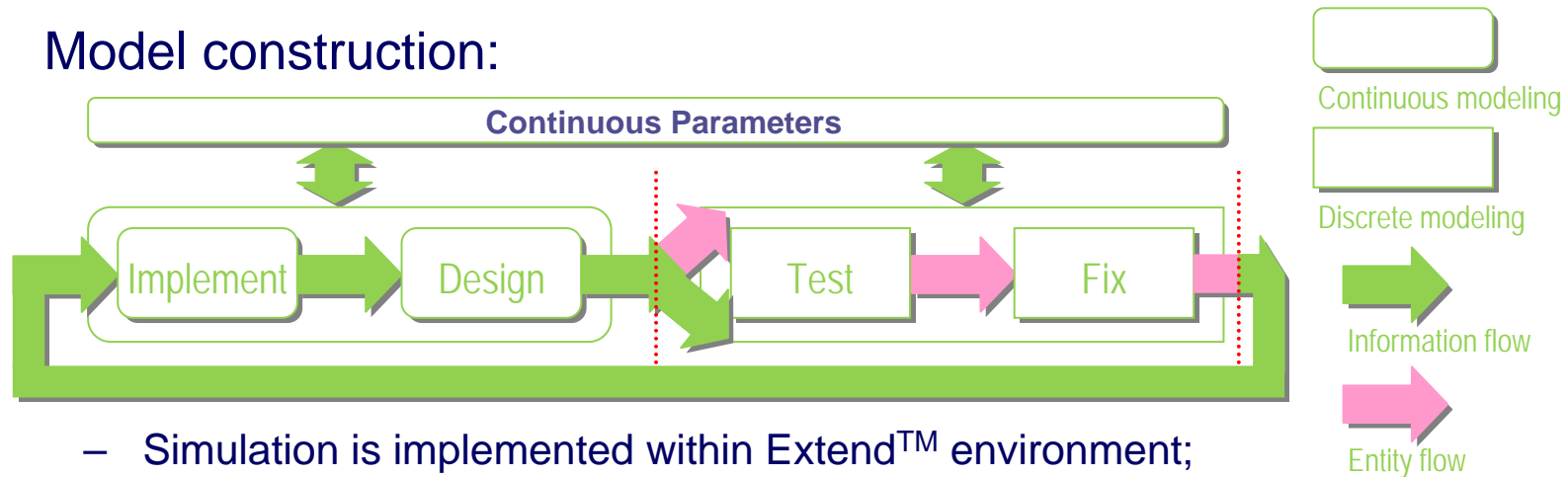
- Approach for module: discrete event simulation
- Basic assumptions:
 - Black-box testing strategy applied;
 - Test suite (including test cases) won't change during process;
 - Test is performed by a standalone team, while defects are fixed by the development team.
- Defect level in test-and-fix:
 - New defects generated in current increment have a higher possibility to be detected than residual defects;
 - Performance of test-and-fix is also determined by defect severity, defect's original developer & release, developer's competency, and process policy.

- Defect sinks during test-and-fix:
 - undetected defects + (detected defects – fixed defects) + bad fixes



defect-fixing process with three developers

- Model construction:



- Simulation is implemented within Extend™ environment;
- ModL language controls the transition and iteration.

- Flows in simulation:

- Information flow: defect level, effort, elapsed time;
entity flow: defect and developer.
- Interface: conversion between entity flow and information flow.

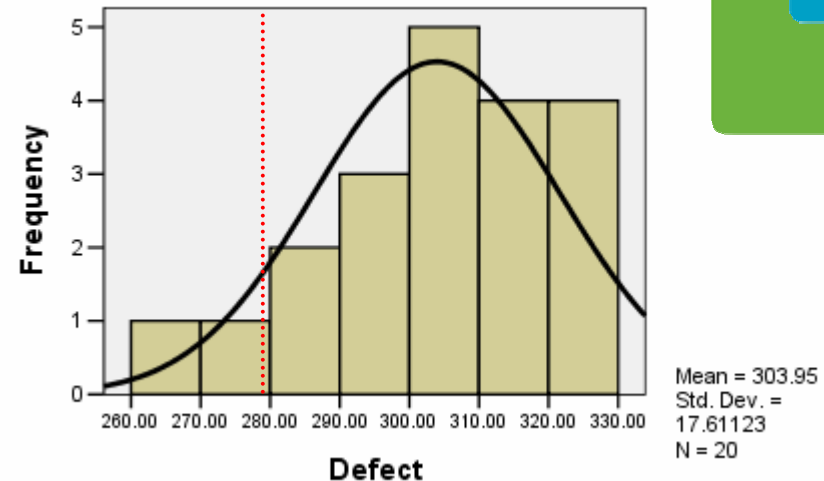
Case study: baseline project



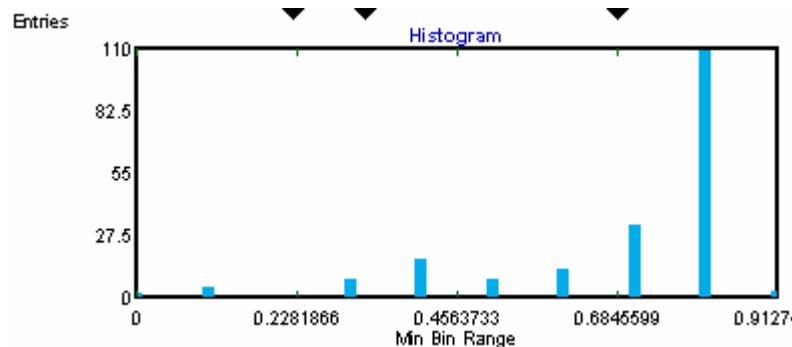
- The baseline project:
 - derived from data reported by Tvedt.

Attributes	Values
Project size	90,667Loc
Number of increments	3
Increment 1	26,667Loc
Increment 2	32,000Loc
Increment 3	32,000Loc
Project schedule	250days (1 year)
Project team size	15 experienced engineers
Estimated budget	3750 man-days

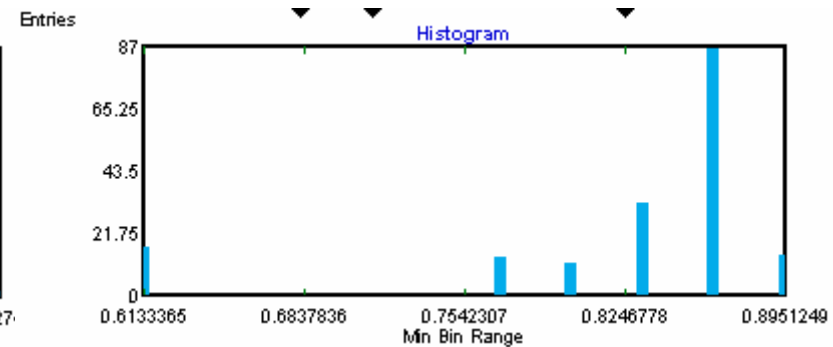
- Estimated product quality:
 - It was reported that 279 defects remains in final product;
 - 20 runs of simulation.



- Process change in test-and-fix phase:
 - Original process: detected defect has to be fixed by its original developer.
 - Defect queue determines workload, some have long queue while some are idle.
 - It becomes severe when competency deviations exist in team.
 - Changed process: defect is assigned to one developer with the shortest defect queue.



developer utility in the original process



developer utility in the changed process

- Conclusion
 - We proposed a horizontal integration scheme for constructing a hybrid process simulation model.
 - Its enhancement lies in flexibly choosing continuous or discrete-event modeling at phase (module) level, and supporting module assembly.
 - We demonstrated the use of horizontal integration by developing a hybrid model of incremental development with focus on test-and-fix process.
- Future work
 - Improve horizontal integration scheme and develop hybrid simulation models of other software processes (e.g. agile process) using this scheme.
 - Enhance this hybrid process model by considering more interventions that are not included at current stage (such as increment dependency).

Questions



Thank you!

Contact:

He (Jason) Zhang
he.zhang@nicta.com.au

Ross Jeffery
ross.jeffery@nicta.com.au

Liming Zhu
liming.zhu@nicta.com.au