

# Quantitative Managing Defects for Iterative Projects: An Industrial Experience Report in China

***Lang Gou<sup>1</sup>, Qing Wang<sup>1</sup>, Jun Yuan<sup>3</sup>, Ye Yang<sup>1</sup>, Mingshu Li<sup>1</sup>, Nan Jiang<sup>1</sup>***

*<sup>1</sup> Laboratory for Internet Software Technologies  
Institute of Software , Chinese Academy of Sciences*

*<sup>3</sup> Beijing ZZNode Technologies Development Co. Ltd.*

# Agenda

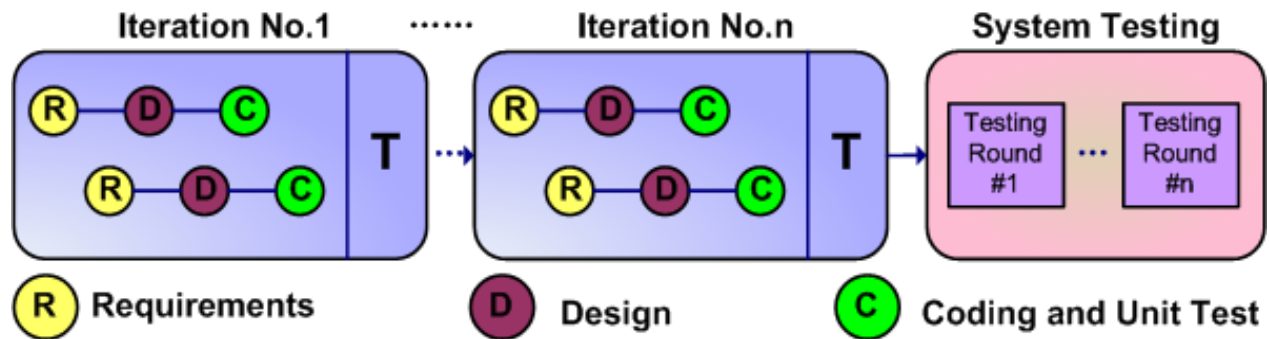
- *Motivation*
- BiDefect Method
- Experience Result of BiDefect
- Benefits, Challenges and Future Work

# Motivation

- Iterative development
  - Avoid a single-pass sequential, gated-step approach
  - More difficult to implement quantitative management
- Quantitative defect management
  - Key to ensure the production of high quality software
  - How to quantitatively manage defects across multiple iterations is a challenging issue.

# Motivation

- Challenges of defect management in iterative development

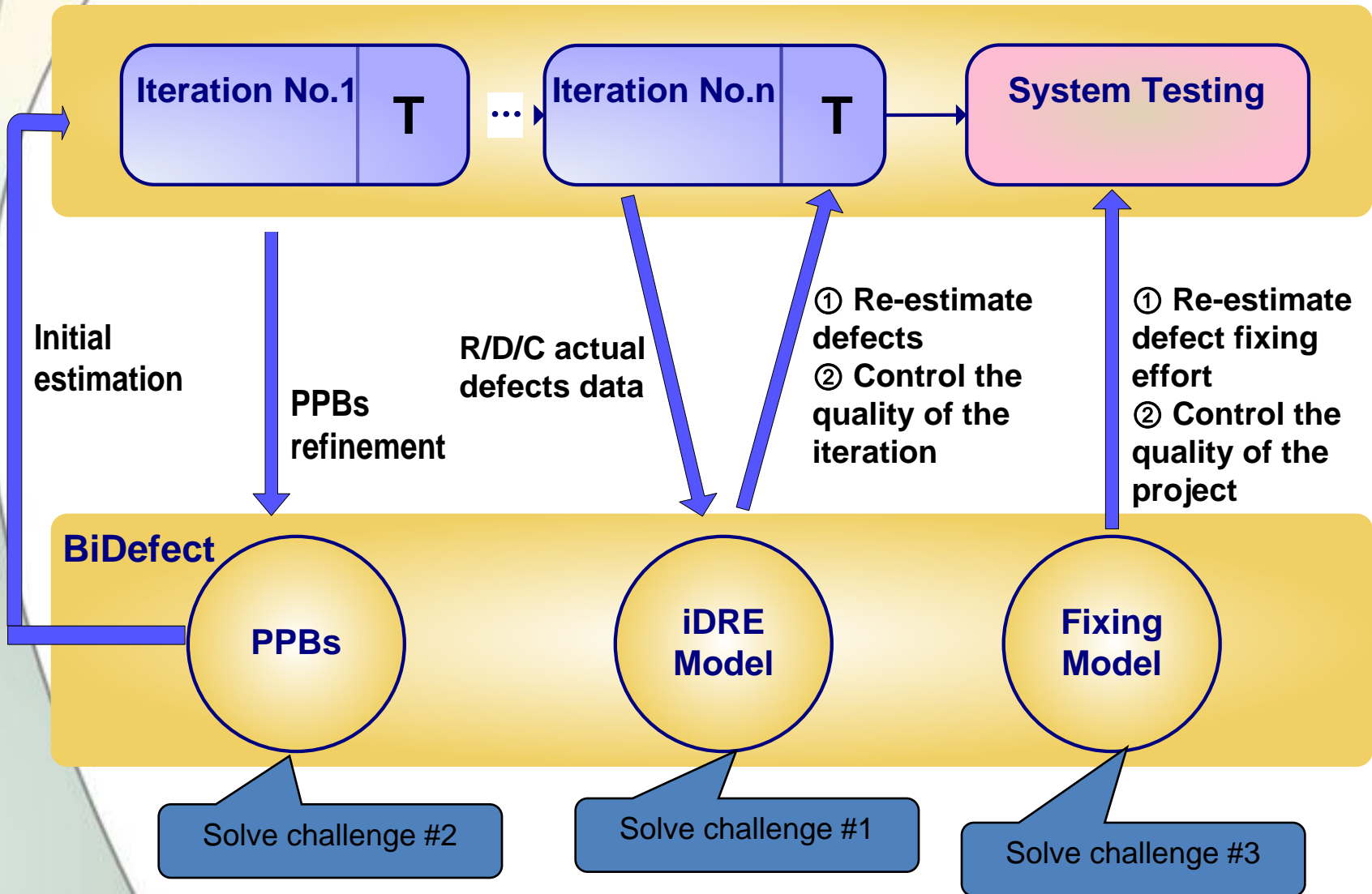


- Challenge 1
  - identifying appropriate “control points” in each iteration.
- Challenge 2
  - selecting appropriate measures and effective measurement methods
- Challenge 3
  - determining the “sweet spot” amount of effort for performing testing and defect fixing activities.

# Agenda

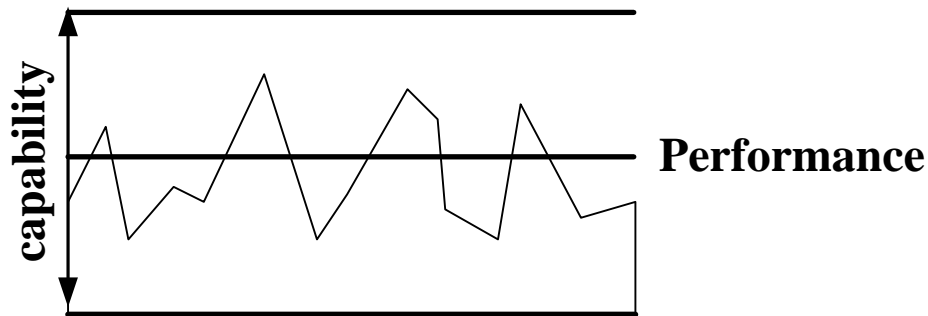
- Motivation
- *BiDefect Method*
- Experience Result of BiDefect
- Benefits, Challenges and Future Work

# The BiDefect Method



# BiDefect part 1: PPBs

- PPBs (*P*rocess-*P*erformance *B*aselines)



- Using BSR[1] method to establish PPBs
  - UCL (Upper Control Limits)
  - CL (Central Line)
  - LCL (Lower Control Limits)

[1]Wang, Q., Jiang, N., Gou L., *BSR: A Statistic-based Approach for Establishing and Refining Software Process Performance Baseline*, ICSE2006

# BiDefect part 1 --- PPBs

- Measures for iterative development projects

No	Measures
1	<b>Defect Injection Rate (DIR)</b> of requirements, design, coding and testing activities = defects injected at the activity / all defects of the project
2	<b>Defect Removal Effectiveness (DRE)</b> of requirements, design, coding and testing activities = defects removed at the activity / (defects existing on activity entry + defects injected during development of the activity)
3	<b>Pre-release defect density</b> = defects removed before product release / product size
4	<b>Post-release defect density</b> = defects detected within one year after product release / product size
5	<b>Productivity</b> = product size / total effort of project
6	<b>Defect Injection Distribution (DID)</b> = defects injected in requirements (or design, coding and testing) / all defects removed in system testing * 100%
7	<b>Percentage of Detecting Effort (PDE)</b> = effort of defect detecting activity in system testing stage / total effort of project * 100%
8	<b>Percentage of Fixing Effort (PFE)</b> = effort of defect fixing activity in system testing stage / total effort of project * 100%
9	<b>Test efficiency</b> = number of defects / defect detecting effort
10	<b>Rework efficiency</b> = number of defects / defect fixing effort

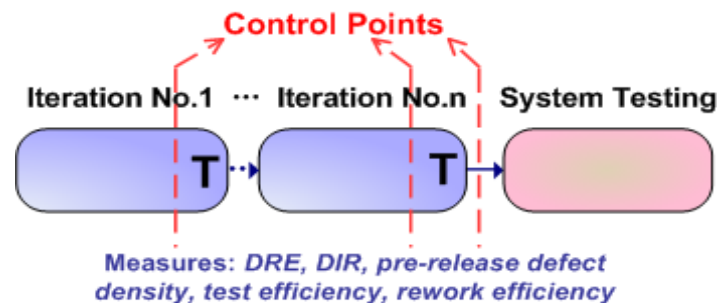
# BiDefect part 2 --- iDRE model

*iteration Defect Removal Effectiveness*

- DRE (**D**efect **R**emoval **E**ffectiveness)
  - summarizes the relationships among three metrics: defect injection, defect removal, and effectiveness
  - For waterfall development, we analyze the DRE of each phase.
  - However, there are not clear phases of requirements, design, coding in an iteration and the DRE cannot be analyzed as in waterfall development.

# BiDefect part 2 --- iDRE model

- iDRE (*i*teration *D*efect *R*emoval *E*ffectiveness)



- Step 1: At the beginning of projects
  - Estimate the defects that could be removed in different kinds of activity of each iteration.
- Step 2: Before the integration testing activities of each iteration
  - Re-estimate the number of defects that could be removed in integration testing
- Step 3: Before the system testing phase
  - Re-estimate the number of defects that could be removed in system testing phase, as well as the defect detecting effort and the defect fixing effort

# BiDefect part 3 --- Fixing model

*PFE: Percentage of Fixing Effort*  
*DID: Defect Injection Distribution*  
*(requirement, design and coding phases)*

- Theory

- Correlation between defects injection distribution and defect fixing effort [1]
- Regression equation

$$PFE = A * DID_{Requirements} + B * DID_{Design} + C * DID_{Coding} + D$$

- Application

- re-estimate the effort of defect fixing based on the actual defects.

[1] Wang, Q., Gou, L., Jiang, N., et al.: An Empirical Study on Establishing Quantitative Management Model for Testing Process. ICSP2007

# Agenda

- Motivation
- BiDefect Method
- *Experience Result of BiDefect*
- Benefits, Challenges and Future Work

# Experience Result

- BiDefect have been successfully applied in ZZNode.
- ZZNode
  - A leading Chinese telecommunications company
  - Founded in 1999
  - about 360 employees
  - 40% product lines adopted iterative development and used BiDefect
  - Aug 15, 2007, CMMI ML 4

# Establish Model

Establish  
PPB for  
each  
measure

PPBs		UCL	LCL	CL
DIR (%)	Requirements	64.1%	8.6%	36.4%
	Design	20.2%	8.0%	14.1%
	Coding	76.0%	13.6%	44.8%
	Testing	9.2%	0.2%	4.7%
DRE (%)	Requirements	85.9%	25.6%	55.8%
	Design	93.9%	2.4%	48.1%
	Coding	52.0%	0.0%	25.5%
	Testing	90.9%	28.0%	59.5%
Pre-release defect density (defects / KLOC)		4.07	3.40	3.70
Post-releases defect density (defects / KLOC)		1.02	0.99	1.01
Software productivity (LOC / Labor Day)		69.20	25.20	47.20
DID (%)	Requirements	9.7%	5.2%	7.4%
	Design	12.2%	5.6%	8.9%
	Coding	80.4%	61.0%	70.7%
	Testing	15.9%	10.1%	13.0%
PDE (%)		2.4%	0.8%	1.6%
PFE (%)		1.6%	0.4%	1.0%
Test efficiency (defects / labor day)		5.59	0.00	2.29
Rework efficiency (defects / labor day)		5.60	0.00	2.61

# Establish Model

- ZZNode analyzed the historical data and derived the regression equation for Fixing model:
  - $$\text{PFE} = 0.1065 * \text{DID\_Requirement} + (-0.0043) * \text{DID\_Design} + (-0.3925) * \text{DID\_Coding} + 0.3597$$
- *Evaluated using F test*
  - At the confidence level  $\alpha=0.05$ , the regression equation was linearly prominent.

*PFE: Percentage of Fixing Effort  
DID: Defect Injection Distribution  
(requirement, design and coding phases)*

# Step1: Initial Estimation from PPBs

- In ZZNode, several iterative projects applied the BiDefect method.
- Brief information about the project

<b># of staff</b>	34 (each staff just involves some activities)
<b>Plan Schedule</b>	11 months
<b>Pan Size</b>	224.6 KLOC
<b>Total Effort</b>	5111 labor days
<b>Lifecycle</b>	Iterative development (total <i>four iterations</i> in overlap time, and <i>three testing rounds</i> in system testing activity)
<b>Techniques</b>	J2EE
<b>Quality Objectives</b>	Post-release defect density $\leq 0.65$ defects/KLOC
<b>Description</b>	To develop the inventory management and service assurance function of the next generation network for PCCW, the biggest telecom operator in Hong Kong, P.R. China.

# Step1: Initial Estimation from PPBs

- Estimation for each iteration based on PPBs

Estimation	1 <sup>st</sup> Iteration	2 <sup>nd</sup> Iteration	3 <sup>rd</sup> Iteration	4 <sup>th</sup> Iteration
Quality objectives (post-release defect density)	0.8	0.8	0.8	0.8
Size (KLOC)	117.3	39.1	47.5	20.7
Schedule (Month)	5.9	2.95	2.5	2.5
Total effort (Labor Day)	2480	829.25	968.75	270.75
Total defects removed	346	115	140	61
Defects removed at requirements	88	29	36	16
Defects removed at design	63	21	26	11
Defects removed at coding	67	22	52	12
Defects removed at testing	128	43	27	23

# Step1: Initial Estimation from PPBs

- Estimation for system testing activity
  - Effort distribution of the three testing rounds: 50%, 30%, 20%
  - Defect distribution of the three testing rounds: 50%, 40%, 10%

<b>Estimation</b>	<b>Testing round #1</b>	<b>Testing round #2</b>	<b>Testing round #3</b>
Effort of defect detecting (Labor Day)	40.9	24.5	16.4
Effort of defect fixing (Labor Day)	25.6	15.3	10.2
Total defects removed in system testing	61	49	12

# Step2: Quantitative Defect Management for Each Iteration Using iDRE

- Collect actual defects removed in requirements ,design and coding, re-estimate the defect that should be removed in testing activity

Actual performance data	1st Iteration	2 <sup>nd</sup> Iteration	3rd Iteration	4th Iteration	
Size (KLOC)	113.8	126	167	225	
Schedule (Month)	Defects Removed Requirements Design Coding Testing				
Total effort (Labor Day)	UCL	132	119	132	191
	LCL	39	3	0	59
Total defects removed	410	154	170	41	
Defects removed at requirements	42	normal, 184 defects escaped		7	
Defects removed at design	90	30	38	12	
Defects removed at coding	85	34	44	12	
Defects removed at testing	193	78	73	0	

Defects injected in testing: 20

To satisfy the quality objectives, defects existed after testing <= 91

Re-estimate defects removed at testing:  
 $\geq 184 + 20 - 91 = 113$

# Step2: Quantitative Defect Management for Each Iteration Using iDRE

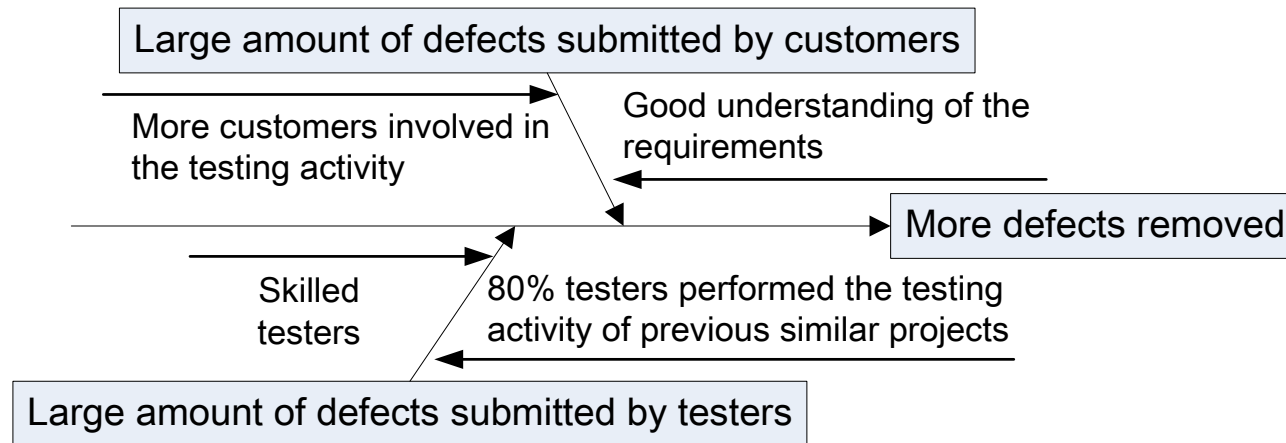
- After the testing activity of the iteration, analyze the abnormality.

Actual performance data	1st Iteration	2 <sup>nd</sup> Iteration	3rd Iteration	4th Iteration
Size (KLOC)	113.8	42.6	46.7	22.5
Schedule (Month)	6	3	2.6	1.5
Total effort (Labor Day)	2512	841	1013.9	289.75
Total defects removed	410	154	170	41
Defects removed at requirements	42	12	15	17
Defects removed at design	90	30	38	12
Defects removed at coding	85	34	44	12
Defects removed at testing	193	78	73	0

More than UCL=191, abnormality!

# Step2: Quantitative Defect Management for Each Iteration Using iDRE

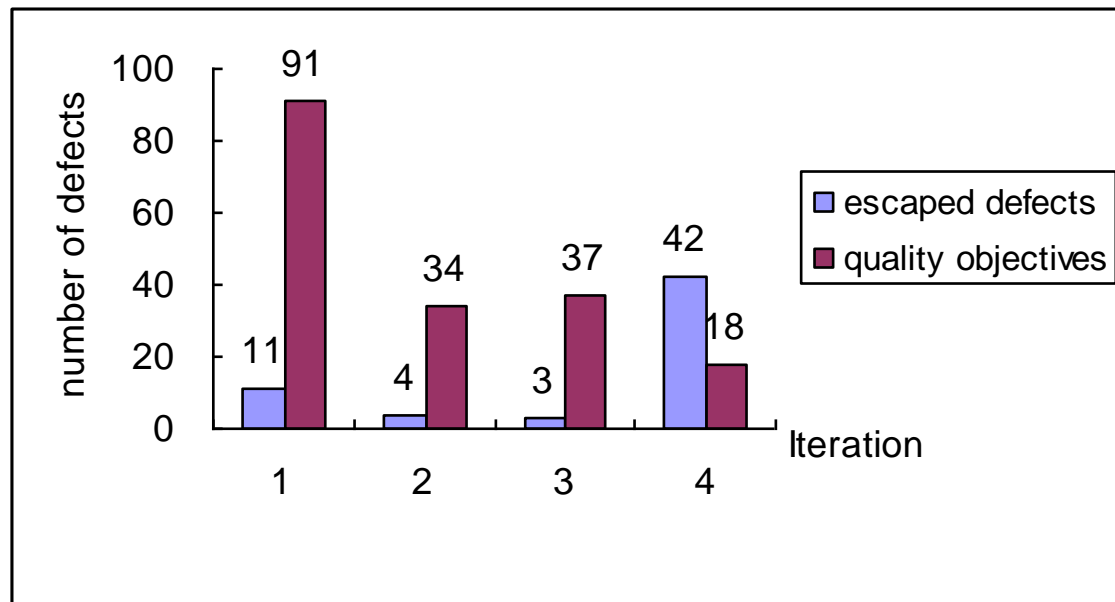
- Causal analysis



- detecting more defects in the first iteration was a good phenomenon and no corrective action was needed.

## Step2: Quantitative Defect Management for Each Iteration Using iDRE

- Satisfaction of quality objectives in each iteration



- The first three iterations all satisfied the quality objectives
- Due to the testing activity in the fourth iteration was moved and integrated with the system testing activity, the fourth iteration did not satisfied the quality objectives.

## Step2: Quantitative Defect Management for Each Iteration Using iDRE

- Re-estimate defects removed in the system testing activity
  - Re-estimation : 73
  - Based on
    - the actual defects removed in the four iterations
    - the quality objectives of the project.
- Re-estimate the effort of testing in system testing activity
  - Based on
    - PPBs of test efficiency

## Step3: Quantitative Defect and Fixing Management for System Testing Using Fixing

- Re-estimation of defects removed in each testing round
  - Based on
    - Re-estimation of defects removed in the system testing activity (73) using iDRE
    - Defect distribution of the three testing rounds: 50%, 40%, 10%
    - Re-estimate defects removed in each testing round: 36, 30, 7
- Re-estimation of defect fixing effort in each testing round
  - Testing round #1
    - Based on
      - Actual defects injected in requirements, design, coding and testing: 3, 3, 26, 5
    - Using regression equation
    - Re-estimate the effort of defect fixing: 26.53 labor days

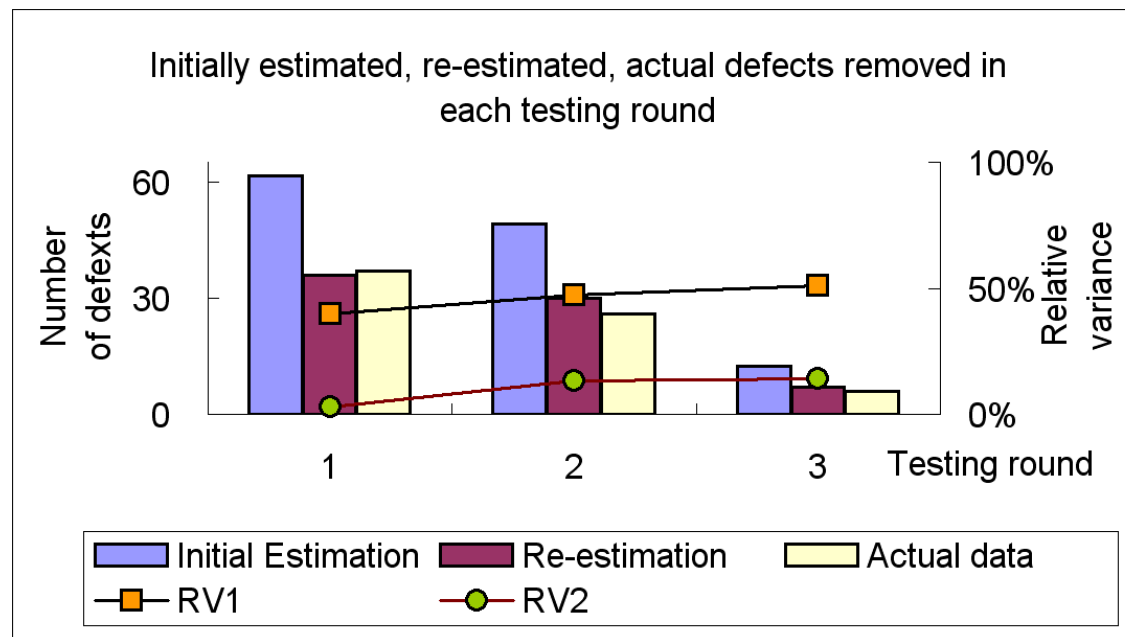
## Step3: Quantitative Defect and Fixing Management for System Testing Using Fixing

- Re-estimated and actual performance of each testing round in system testing process

Process-performance	Testing round #1		Testing round #2		Testing round #3	
	Re-estimation	Actual data	Re-estimation	Actual data	Re-estimation	Actual data
Effort of defect detecting (Labor Day)	41.2	22.75	28.8	11.75	12.4	7.25
Effort of defect fixing (Labor Day)	26.53	28	26.98	21	13.09	10
Total defects removed in system testing	36	37	30	26	7	6
Defects injected in requirements	3	3	2	1	0	0
Defects injected in design	3	3	3	2	1	1
Defects injected in coding	25	26	21	17	5	4
Defects injected in testing	5	5	4	6	1	1

## Step3: Quantitative Defect and Fixing Management for System Testing Using Fixing

- Comparison of initial estimation, re-estimation and actual performance
  - $RV1 = \frac{ABS(\text{actual data} - \text{initial estimation})}{\text{initial estimation}} * 100\%$
  - $RV2 = \frac{ABS(\text{actual data} - \text{re-estimation})}{\text{re-estimation}} * 100\%$



- the re-estimated defects were more precise than the initial estimation since it is closer to the actual defects

# Agenda

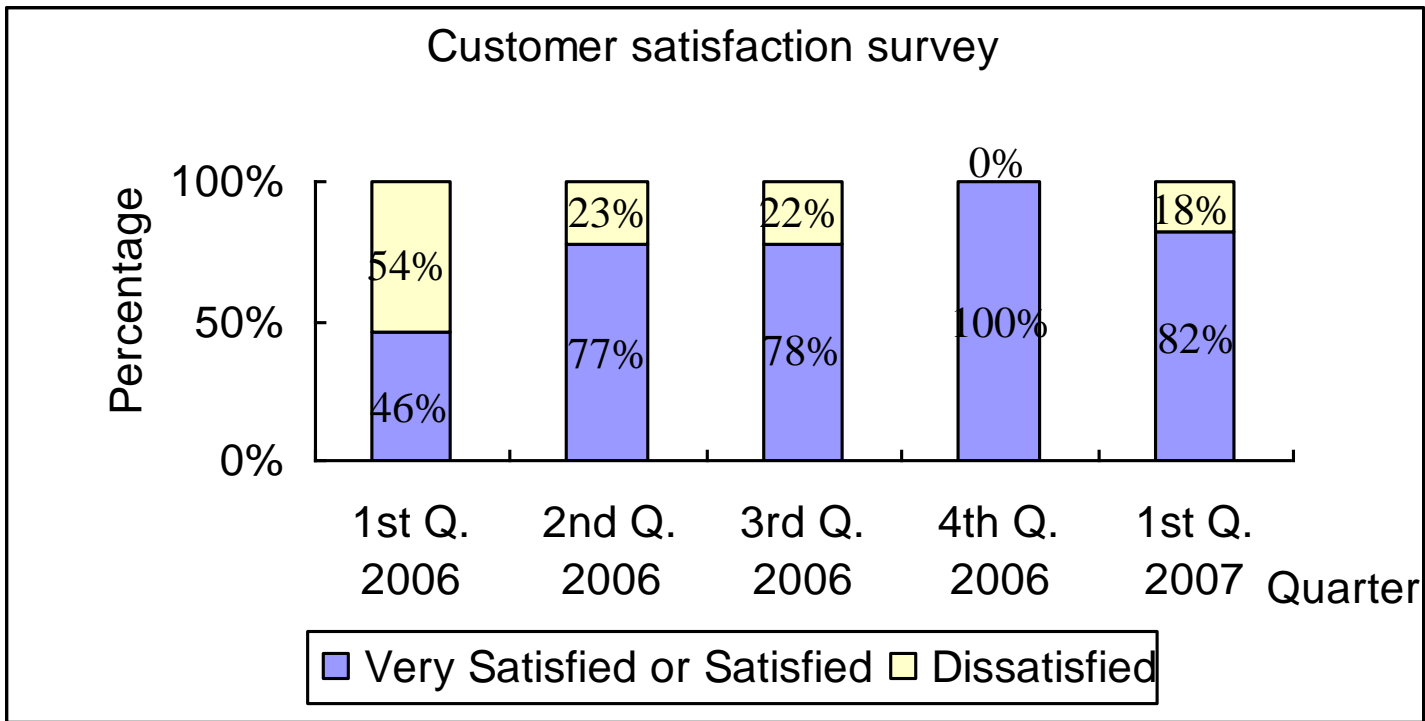
- Motivation
- BiDefect Method
- Experience Result of BiDefect
- *Benefits, Challenges and Future Work*

# Benefits of using BiDefect 1

- From ZZNode
  - Better management of defect data repository.
    - The PPBs (Process Performance Baselines) were established according to organization's defect management objectives. Compared to the previous PPBs, current PPBs are more appropriate
  - Quantitatively controlling projects efficiently
    - The BiDefect method was coherent to ZZNode's business objectives, and covered the entire lifecycle

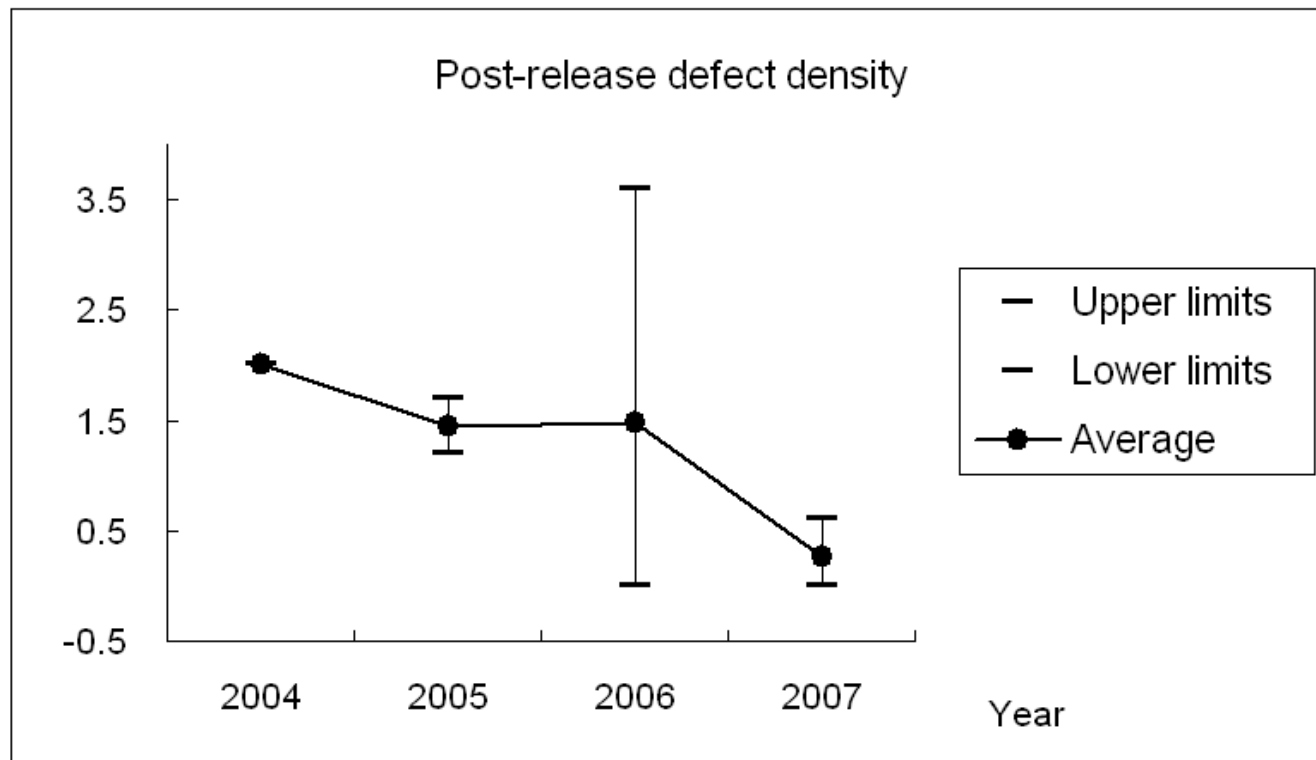
# Benefits of using BiDefect 2

- From ZZNode
  - Promotion of customer satisfaction.



# Benefits of using BiDefect 3

- From ZZNode
  - Improvement of product quality



# Challenges and Lessons Learned

- Dependent on quality of data
  - The results of applying BiDefect are most dependent on the availability and quality of historical data in the organization who use them.
  - If the historical data is missing or not sufficient, we recommend collecting more data from stable process before using the BiDefect method

# Future Work

- Extend BiDefect to suit more generalized iterative process needs;
- Integrate with SoftPM and automate the defect management process;
- Validate and improve BiDefect in other selected organizations.

A large, decorative graphic on the left side of the slide. It consists of several overlapping, curved shapes in shades of light blue, yellow, and green, creating a sense of movement and depth. The shapes are semi-transparent and overlap each other, with the yellow shape being the most prominent in the upper left.

# Thank You!