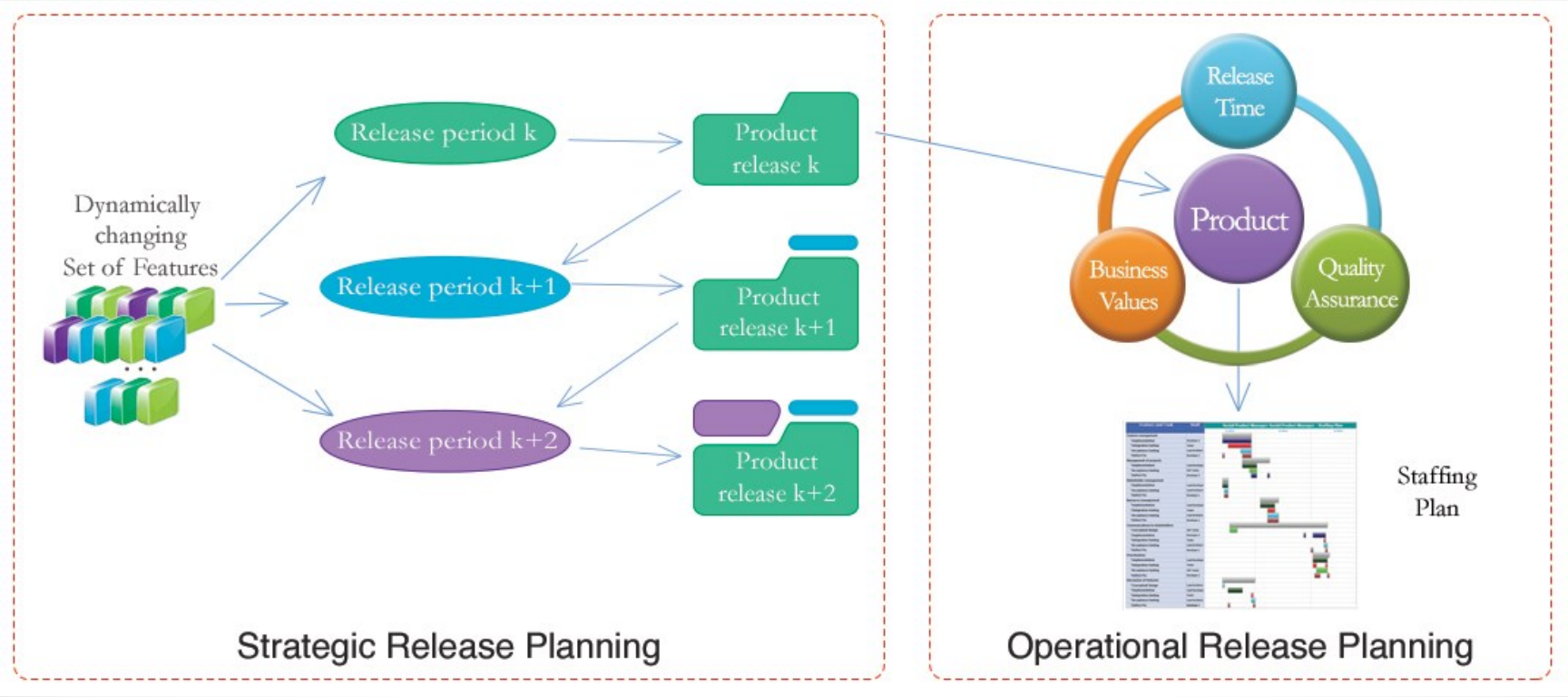
Releasing sooner or later?

Jason Ho, Guenther Ruhe
University of Calgary

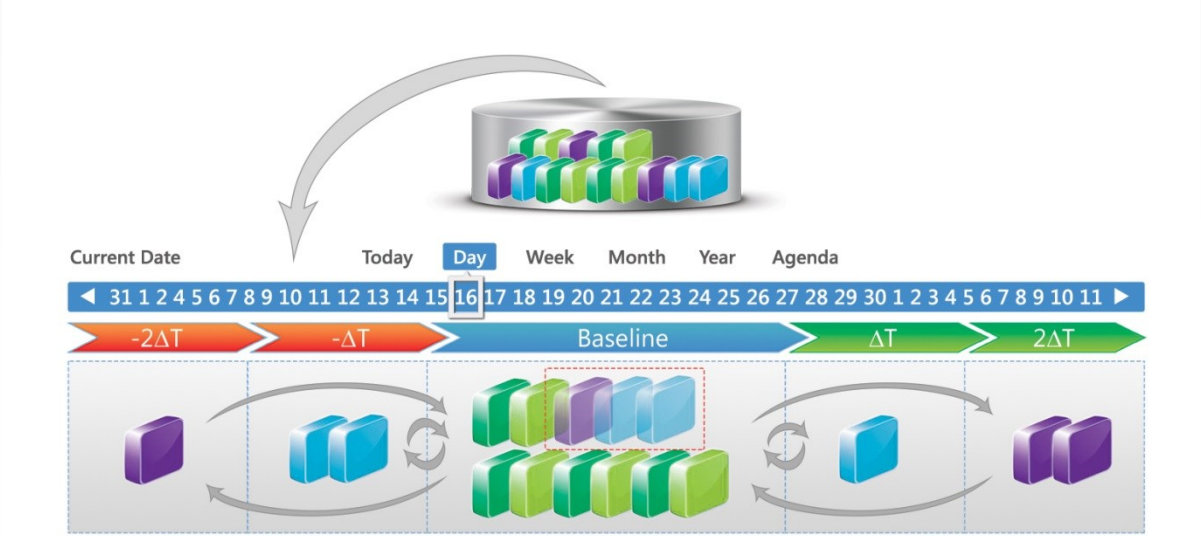
Outline

- Problem Definiton
  - Release Planning Problem
  - When-to-release Problem (W2RP)
  - Research Questions
  - Innovations
- Approach
  - Modeling
  - Effort re-allocation
  - Trade off Solutions
- Evaluation - Case Study
- Limitations & Outlooks
- References
- Q & A

Release Planning Problem



When-to-release (W2RP)



- RQ1: Given a *specific release date*, by varying around a duration, how can we identify an optimized release date?
- RQ2: What is the trade-off between *the value* (stakeholders' satisfaction) and *the assured quality* (reliability) of the release plan?

Contributions

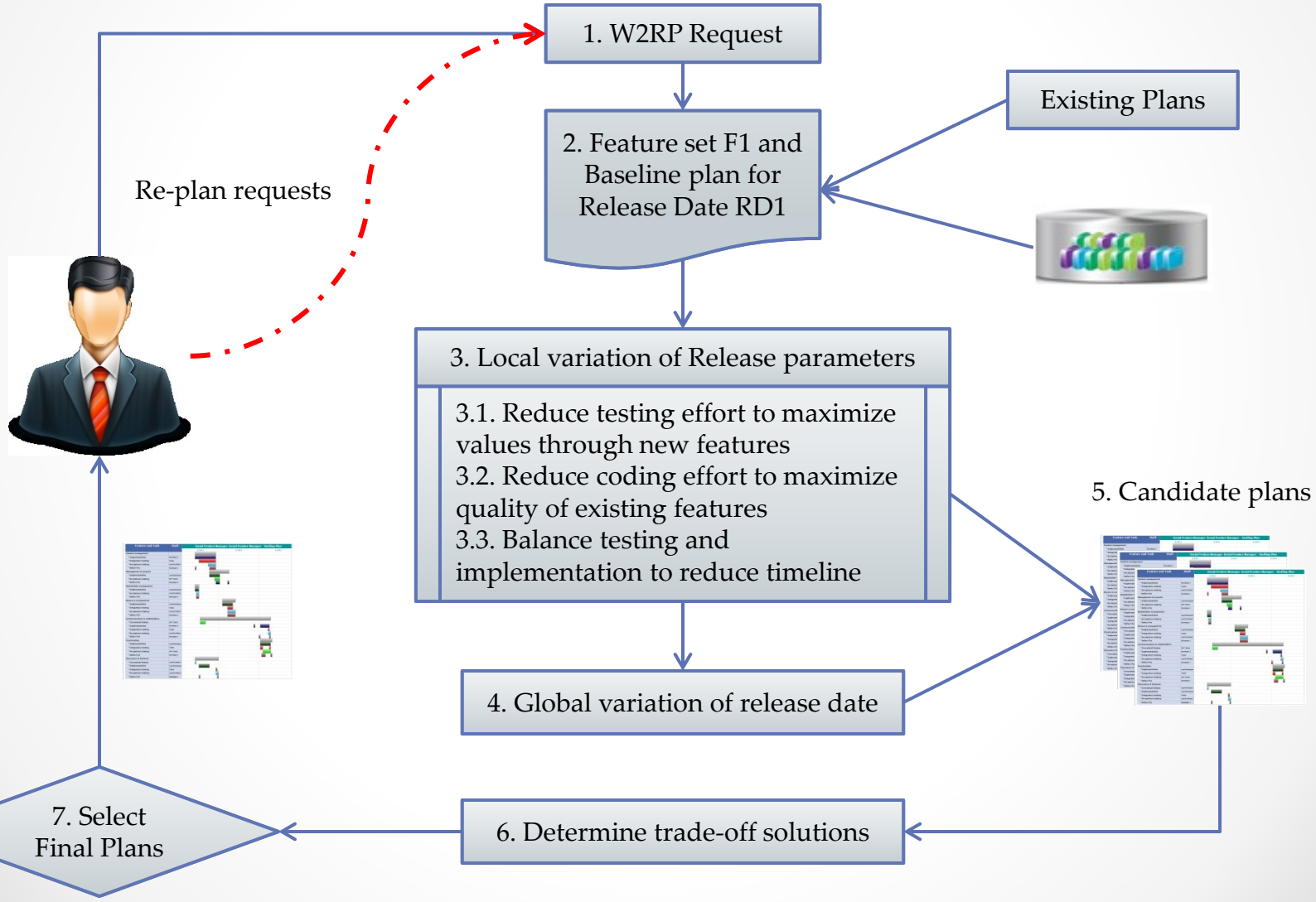
- As an independent tool-plugin:
  - W2RP can be used as-is with existing processes and existing tools
  - W2RP presents instant and interactive what-if solutions
- During Strategic Planning
  - Different alternatives for when-to-release date with predictable outcomes
- During Operational Execution
  - As the project progresses, more defect data will be available → increase in accuracy of the prediction model of quality → Re-planning potential
- Challenges:
  - Complexity of assigning the right resources to the right task at the right time
  - Trade-off between different criteria while maintain quality and value benchmark

Modeling

- Time:
  - RD: *Targeted time* to be released by stakeholders (calendar dates)
  - RD ± ΔT: The duration in which the release date can be varied to find the *optimized release time*
- Values:
  - Measured by Customers' *weighted satisfaction* score
  - As each feature consumes resources, values is affected by capacity of the resources assigned to that feature set.
- Quality:
  - Defined by certainty level of successful transactions after releases
  - Quality is assured by investing effort for testing (Cost of Quality), which comprises of Cost of Conformance (effort for designing test cases) and Cost of Lack of Conformance (effort for fixing bugs)

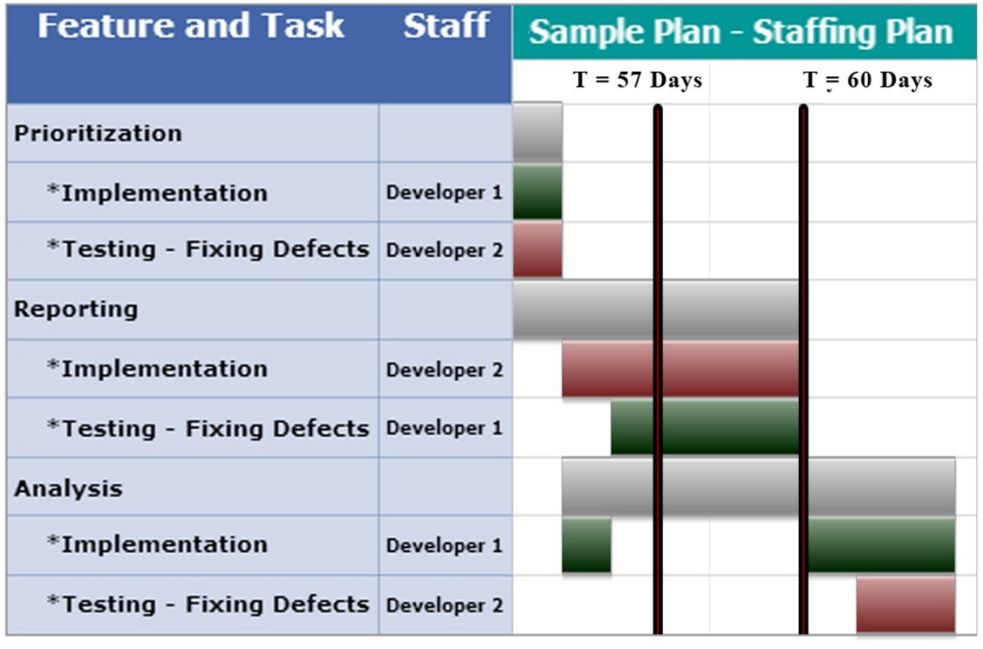
    $$CoQ = CoC + CoLC$$
  - As testing consumes resources, quality is affected by capacity

Approach


1. W2RP Request
Existing Plans
2. Feature set F1 and Baseline plan for Release Date RD1
Re-plan requests
3. Local variation of Release parameters
3.1. Reduce testing effort to maximize values through new features
3.2. Reduce coding effort to maximize quality of existing features
3.3. Balance testing and implementation to reduce timeline
5. Candidate plans
4. Global variation of release date
6. Determine trade-off solutions
7. Select Final Plans

Effort Re-allocation

- Plan stability
  - Changes in timeline will only affect more recent features
- Priority:
  - Effort in building new/important functionalities
  - Effort in testing built/existing functionalities
  - Balanced for the best timeline



Evaluation - Case Study

- We evaluate the approach using a Case study from *a real life* technical project
- Objectives:
  - Evaluate Optimization approach
  - Collect data on potential Trade-off solutions
- Case set up:

Project name	Social Product Manager
Total No of features	40
Features for next release F_0	22
Number of staffs	7
Original release date RD_0	80
Max value $TRV(F_0)$	137

Case-study – Trade-off Solutions

- Potential trade-off solutions
  - Maximize Total Release Values TRV(F_i)
  - Maximize Total Release Quality TRQ(F_i)
  - Minimize Time to release RD_i

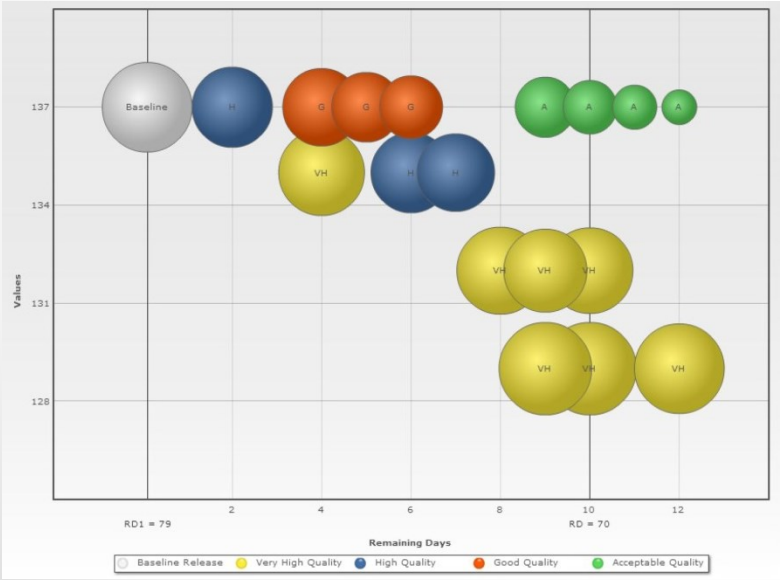


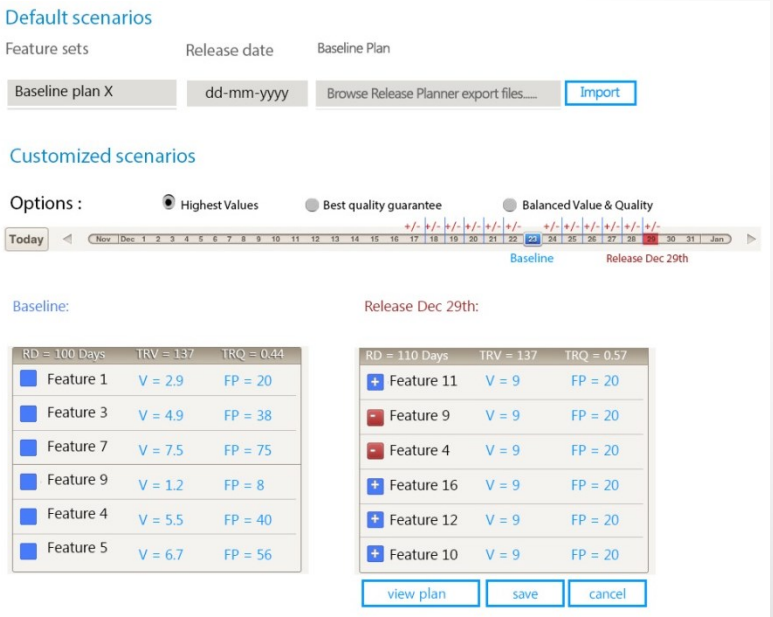
Table: Trade-off solutions for consideration

Trade-off solutions			
ΔT (Days)	RD Duration (Days)	P of ≤5% Defect Rate	Total Release Value
12	68	0.057	137
12	68	0.040	129
11	69	0.044	132
11	69	0.055	137
10	70	0.053	137
10	70	0.042	132
10	70	0.039	129
9	71	0.051	137
9	71	0.043	132
9	71	0.039	129
8	72	0.041	132
7	73	0.045	135
6	74	0.044	135
6	74	0.050	137
5	75	0.048	137
4	76	0.045	137
4	76	0.042	135
3	77	0.043	135
2	78	0.044	137

Limitations & Outlook

- Limitations
  - Reliant on extensive data of number of test cases, defect rate, and fix rate which may not be well-defined in real-life, complex projects.
  - Do not consider fixing and revising of requirements and design.

- Future works:
  - Integration to existing tools
  - Consider different optimization approaches for re-allocation
  - Conduct more in-depth analysis and evaluations



References

- [1] B. Boehm, V. R. Basili. "Defect Reduction Top 10 List", Computer (2001), pp. 135-137.
- [2] J. Campanella, "Principles of Quality Costs: Principles, Implementation and Use, "ASQ Quality Press, 1999.
- [3] J. M. Elroy, G. Ruhe, "When-to-release decisions for features with time-dependent value functions", *Requirements engineering* 15.3 337-358, 2010.
- [4] B.H. Far "Software Reliability Models", *Dependability & Reliability of Software Systems,* Chapter 6, Lecture Notes, 2012.
- [5] Polat, G., Arditi, D., & Mungen, U. "Simulation-based decision support system for economical supply chain management of rebar". *Journal of construction engineering and management*, 133(1), 29-39, 2007.
- [6] J. Ho, G. Ruhe, "Releasing sooner or later: An optimization approach and its case study evaluation", Unpublished, IEEE, 2013.
- [7] R. Lai, G. Mohit, and P. K. Kapur. "A Study of When to Release a Software Product from the Perspective of Software Reliability Models." *Journal of Software* 6 (2011), pp. 651-661.
- [8] M. Przepiora, "A Hybrid Release Planning Method for Accommodating Advanced Feature Dependencies", Department Of Computer Science, University of Calgary (Master Thesis), 2012.
- [9] G. Ruhe, "Product Release Planning: Methods, Tools and Applications", Calgary, AB, Canada: CRC Press, Chapter 8.3, pp. 157-160, & Chapter 10.4, pp. 221-227, 2010.
- [10] L. Zawadzki & T. Orlova, "Building and Using a Defect Prediction Model". Chicago Software Process Improvement Network, Feb, 2012.