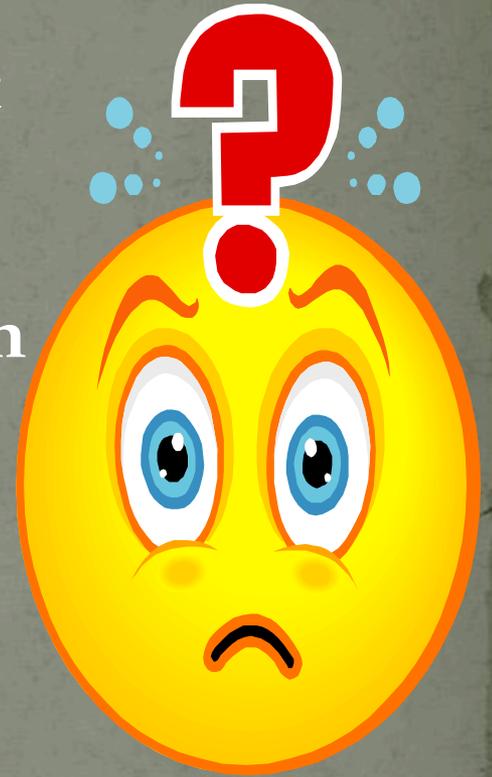


A System Dynamics
Approach for Effective
Delivery of Agile Software
Projects

By Olumide Akerele

What is Continuous Delivery

- Continuous Delivery is the improvement in the delivery process of software, with strong emphasis on **test automation, continuous integration, collaboration and deployment automation**. The ultimate goal of CD is to release good quality software effortlessly to the customer on a frequent basis.



Rationale for Continuous Delivery?

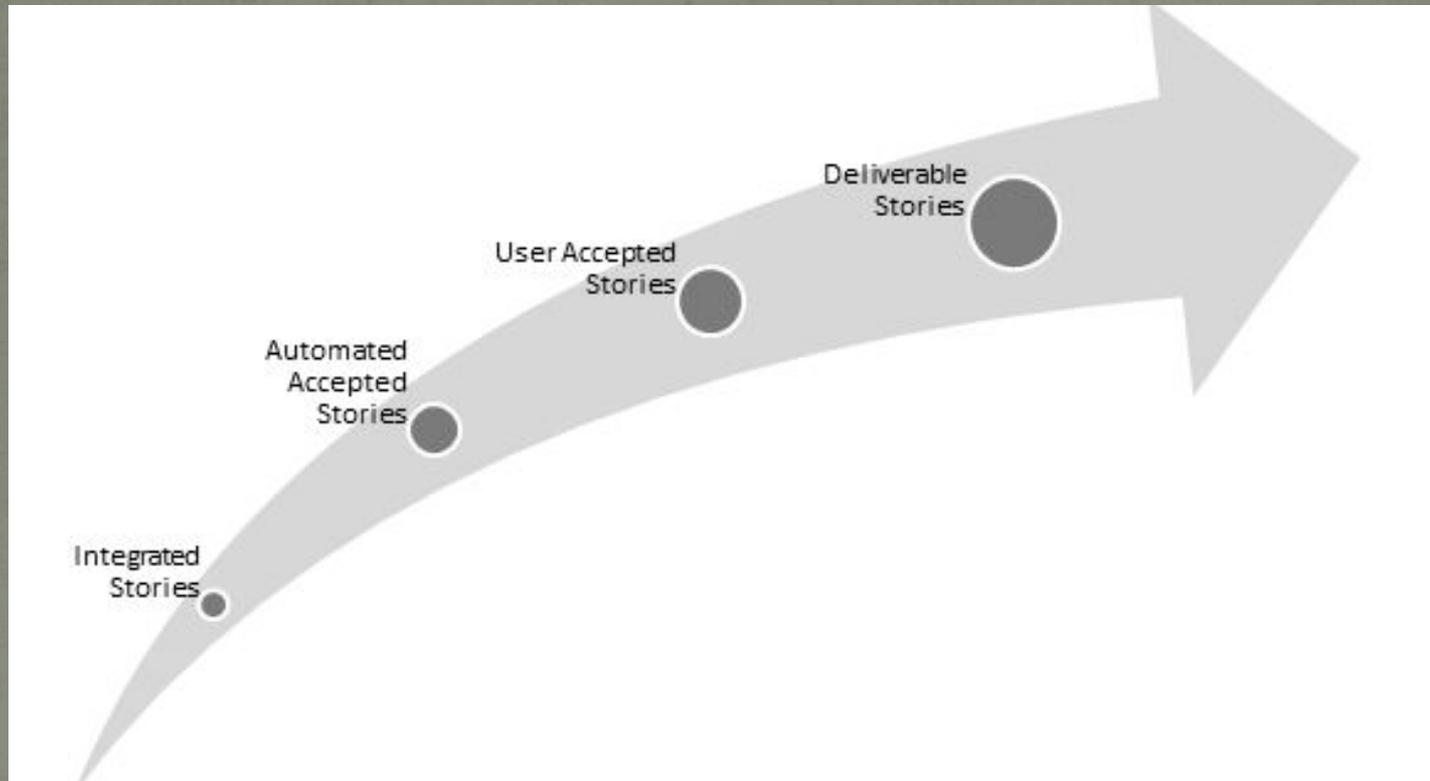
- Stressful manual releases.
- Over-reliance on 'experts' before deployment.
- Lack of process repeatability
- 'waste' of skilled labor for monotonous activities
- The slowness of manual deployment means it's no good match with agile development whose major propriety is on frequent delivery



CD Core practices

- Continuous Integration: Each developer commits to trunk(mainline), runs comprehensive unit tests.
- Test Automation: Unit tests, functional tests, smoke tests.
- Automated Deployment: pull system, ensure all configurations (OS, middleware, infrastructure, application)in version control.
- Improved Collaboration: breaking down 'silos'.

CD Pipeline Artefacts



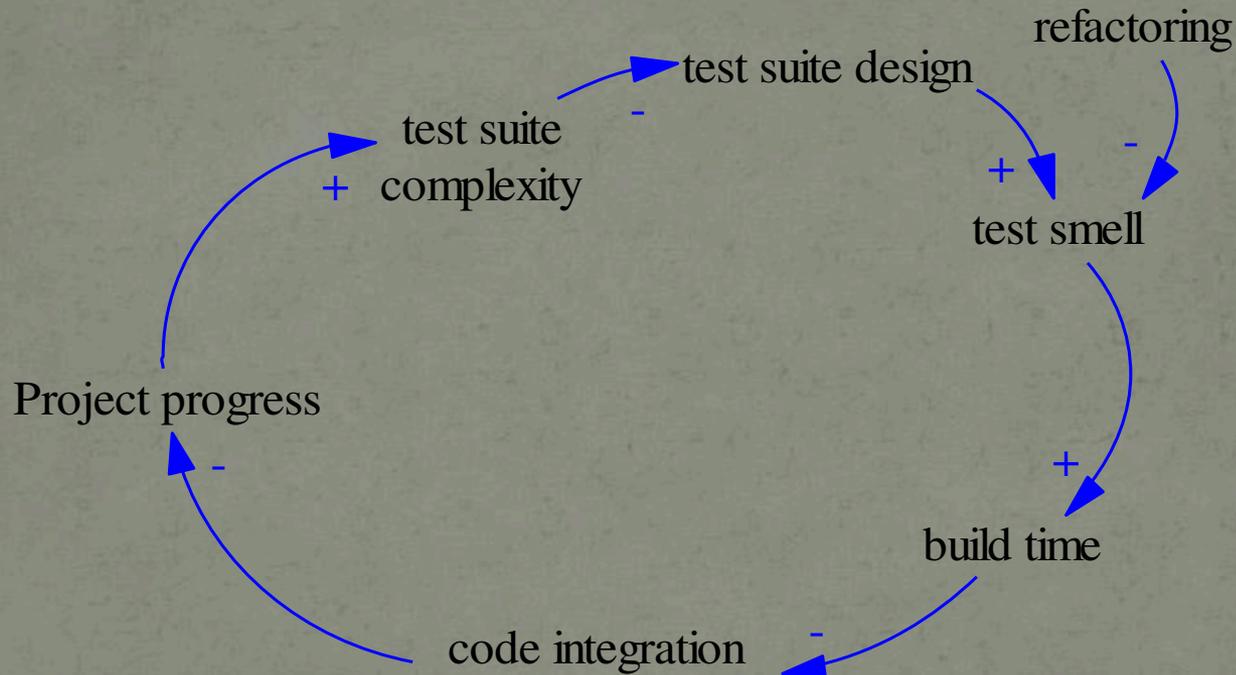
Dynamics in Software Projects

“no one thing seems to cause the difficulty (in software projects).... but the accumulation of simultaneous and interacting factors...”- **Brooks**

- interactions and interdependencies are prominent in software development due to environmental, human and technological factors- and these are accountable for the inconsistencies in software project results .
- Factors such as level of developer experience, team culture, schedule pressure, learning curve of tools, perceived complexity of software projects..... are vital factors that determine project performance.



Dynamics in Automated Acceptance Testing



Resulting Problems

- Unexpected outcome
- Schedule delay
- Inconsistent quality
- Varying cost
- Poor planning precision
- Poor customer satisfaction



RQs

- What are the post development variables in software projects that have a significant impact on the continuous delivery of software .
- What are the agile practices that have an impact on the CD process?
- As automation of tests and builds is a major constituent of CD and the build duration is dependent on the number and complexity of tests , what are the responses of developers to the duration of the build?
- What is the impact of refactoring automated acceptance test suite?

Research Objectives

- Investigate all the factors that have an impact on the success determining practices of continuous delivery.
- Study the full dynamics of these factors and relevant agile practices on the continuous delivery process.
- Design the system dynamic model for continuous delivery to provide a high level insight of “actions and reactions” within the CD context.
- Run simulation and compare results for validation.
- Model exploration/experimentation.

Methodology

- Interviews
- Literature reviews
- Surveys
- Project Historical Data

Approach: System Dynamics (SD)

- SD is used to model dynamic problems arising in complex social, managerial, economic, or ecological systems - literally any dynamic systems characterized by interdependence, mutual interaction, information feedback, and circular causality.
- It models the various relationships between elements of a particular system and how these relationships influence the behaviour of the system over time.

Why System Dynamics?

- Clear representation of causal effects of variables within the system and the complexity of their inter-relationship - ".... the feedback structures of real problems are often so complex that the behavior they generate over time can usually be traced only by simulation" - Madachy
- System behaviour easy to analyse overtime.
- Dynamics of system variables are apparent.
- Unfeasibility of empirical methods – SD facilitates easy leeway on variables.

Validation

Two Stages:

- Structural Validation
- Behavioural Validation

Challenges

- Difficulty in getting sufficient aggregated data from case study
- Limitations in the number of Companies that actually release frequently and keep metrics for each release

references

- Humble, J. and Farley, D. *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*. Addison Wesley, 2010.
- A. Lahtela and M. Jantti, “Challenges and problems in release management process: A case study,” in *2011 IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS)*, 2011, pp. 10 –13.
- Borg, R. and Kropp, M. Automated Acceptance Test refactoring. *Proceedings of the 4th Workshop on Refactoring Tools*, ACM (2011), 15–21.
- Ogata, K. *System Dynamics*. Prentice Hall, 2003.

Thank you!!!

Questions??