

#### Balancing Dev Agility With your IT Ops Mission Using Kanban



#### Dominica DeGrandis @Dominicad

# Cycle of Frustration

#### THINGS TAKE TOO LONG!

#### THERE'S NO VISIBILITY

TOO MUCH WORK!

#### CONFLICTING PRIORITIES

### When everything is a priority 1



#### Context switching is expensive.



#### And irritating!

because ...

- Multitasking "stop & go" breaks the on-a-roll flow.
- Churn and rework create waste.

1. Make work visible

#### Bringing visibility to Ops work

![](_page_5_Figure_1.jpeg)

- It's hard to manage invisible work.
- Humans absorb info faster visually than any other sense.
- Visualization helps us see patterns & connections that matter.

## When the front end gets fixed without addressing the backend.

![](_page_6_Figure_1.jpeg)

Make work visible
Limit work-in-progress

#### Bringing limits to amount of work in play

![](_page_8_Figure_1.jpeg)

- Knowledge work is perishable.
- Churn and rework create waste.
- Increased cycle time increases cost.

### Time weighs heavily on heroes

![](_page_9_Picture_1.jpeg)

- 1. Make work visible
- 2. Limit work-in-progress
- 3. Make process policies explicit

#### Rules contribute to flow

![](_page_11_Picture_1.jpeg)

#### Infrastructure contributes to congestion

![](_page_12_Picture_1.jpeg)

## Consider options that allow for improved flow

![](_page_13_Picture_1.jpeg)

Roundabouts decrease "stop & go" increasing throughput by 20%. What rules would increase Ops throughput?

#### Ops workflow sample rules

![](_page_14_Figure_1.jpeg)

#### Calculating Cost of Delay

#### Consists of the sum of –

- Missed Revenue
  - New revenue
  - Protecting Existing revenue
- Extra costs
  - Running costs (dev staff, ops staff, ops cost)
  - Fines and penalties (regularity, contractual)

- Sum of Cost of Delay of Other Projects (often missed)

Troy Magennis - http://focusedobjective.com/cost-of-delay/

- 1. Make work visible
- 2. Limit work-in-progress
- 3. Make process policies explicit
- 4. Measure flow

#### Measure workflow

![](_page_17_Figure_1.jpeg)

#### because ...

- Historical cycle time provides ability to set customer expectations.
- Discovering where and why work gets bottlenecked is useful.
- Looking at trends over time reveals improvement (or not).

#### How long has work been idle?

![](_page_18_Figure_1.jpeg)

## Rate of incoming vs. outgoing

![](_page_19_Figure_1.jpeg)

## Measure for objectivity

![](_page_20_Picture_1.jpeg)

- 1. Make work visible
- 2. Limit work-in-progress
- 3. Measure flow
- 4. Make process policies explicit
- 5. Develop feedback loops

#### Opportunities to hear and to be heard

natizal

#### Lean Coffeehttp://leancoffee.org/

### **Operations Review**

![](_page_23_Picture_1.jpeg)

#### Reflect on quantitative performance measures

#### Quantitative performance measures

- WIP (work-in-progress)
- Cycle time/Lead time
- Throughput
- Quality
- Issues & blocked work

Across:

## Trend & Variability

- 1. Make work visible
- 2. Limit work-in-progress
- 3. Measure flow
- 4. Make process policies explicit
- 5. Develop feedback loops
- 6. Evolve Experimentally using models& theory

## Experiment with applicable models & theory

![](_page_26_Picture_1.jpeg)

From the authors of The Visible Ops Handbook

![](_page_26_Picture_3.jpeg)

Gene Kim, Kevin Behr, and George Spafford

![](_page_26_Picture_5.jpeg)

Haw Taday's Entropreneurs Use Continuous Innovation to Create Radically Successful Businesses

![](_page_26_Picture_7.jpeg)

Second Edition

Forecasting and Simulating Software Development Projects

Effective Modeling of Kanban & Scrum Projects using Monte-carlo Simulation

![](_page_26_Figure_10.jpeg)

Includes free license to Focused Objective's Scrum and Kanbar Monte-carlo Simulation toolkit.

![](_page_26_Picture_12.jpeg)

Troy Magennis @AgileSimulation

FocusedObjective.com

The Principles of Product Development FLOW Second Generation

Lean Product Development

DONALD G. REINERTSEN

![](_page_26_Picture_17.jpeg)

KANBAN

Successful Evolutionary Change for Your Technology Business

![](_page_26_Picture_20.jpeg)

David J. Anderson

![](_page_27_Picture_0.jpeg)

#### Balance Dev Agility with your IT Ops Mission Using *Kanban*

Ready (6)	envestigate (3)	- Devel Doing	opment (5) Done	Opt Doing	(2) Done	🚑 UAT (3)	Delivered
Expedite (+1)							
Ball and Balls To Ball and Balls To			Marine Bolts 10	The second of th			

<u>www.ddegrandis.com</u> *Kanban for Devops & IT Ops* dominica@ddegrandis.com

Dominica DeGrandis @Dominicad