

poorlydrawnlines.com



# When creating software, what **Really** Matters?

Kevin Trethewey

# When creating software, what **Really** Matters?



## Agenda...

- How I got here
- How you got here
- **6** Things
- Discussion

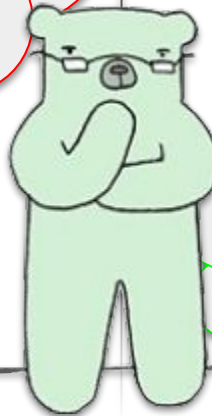
No tertiary  
education

Not very good  
at following  
instructions

Used to be an  
electrician

Hasn't had a  
full time job  
for > 10 years

Completely  
unfamiliar  
with your tech  
stack



Founded &  
Ran a  
software  
company

Has built  
many  
enterprise  
scale systems

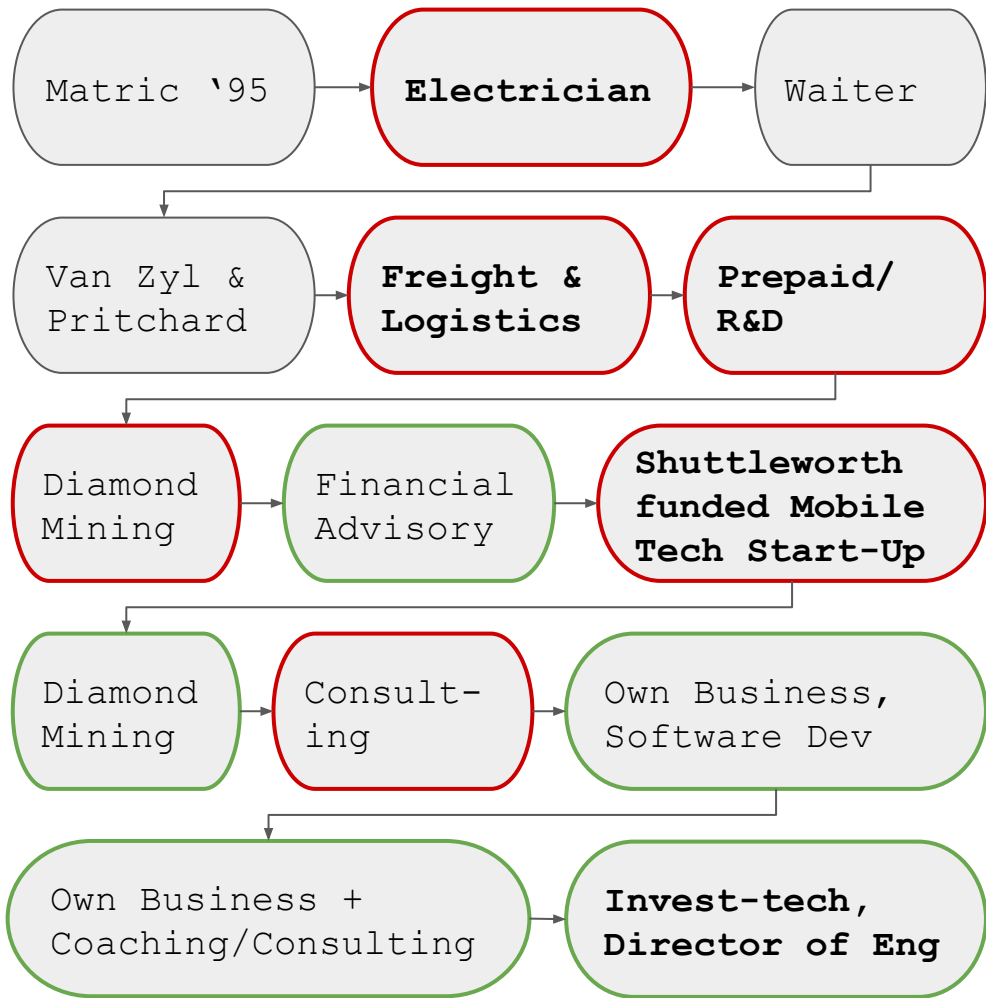
Coached and  
guided  
developers &  
teams

Recognised Agile  
org design  
expertise

Has a strong  
reputation in the  
Dev community



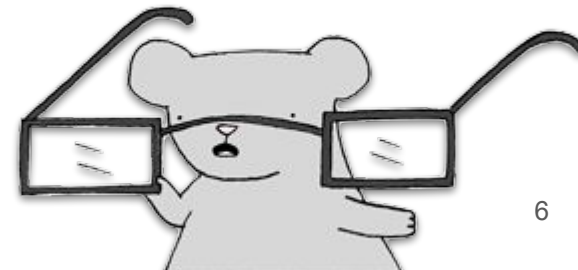
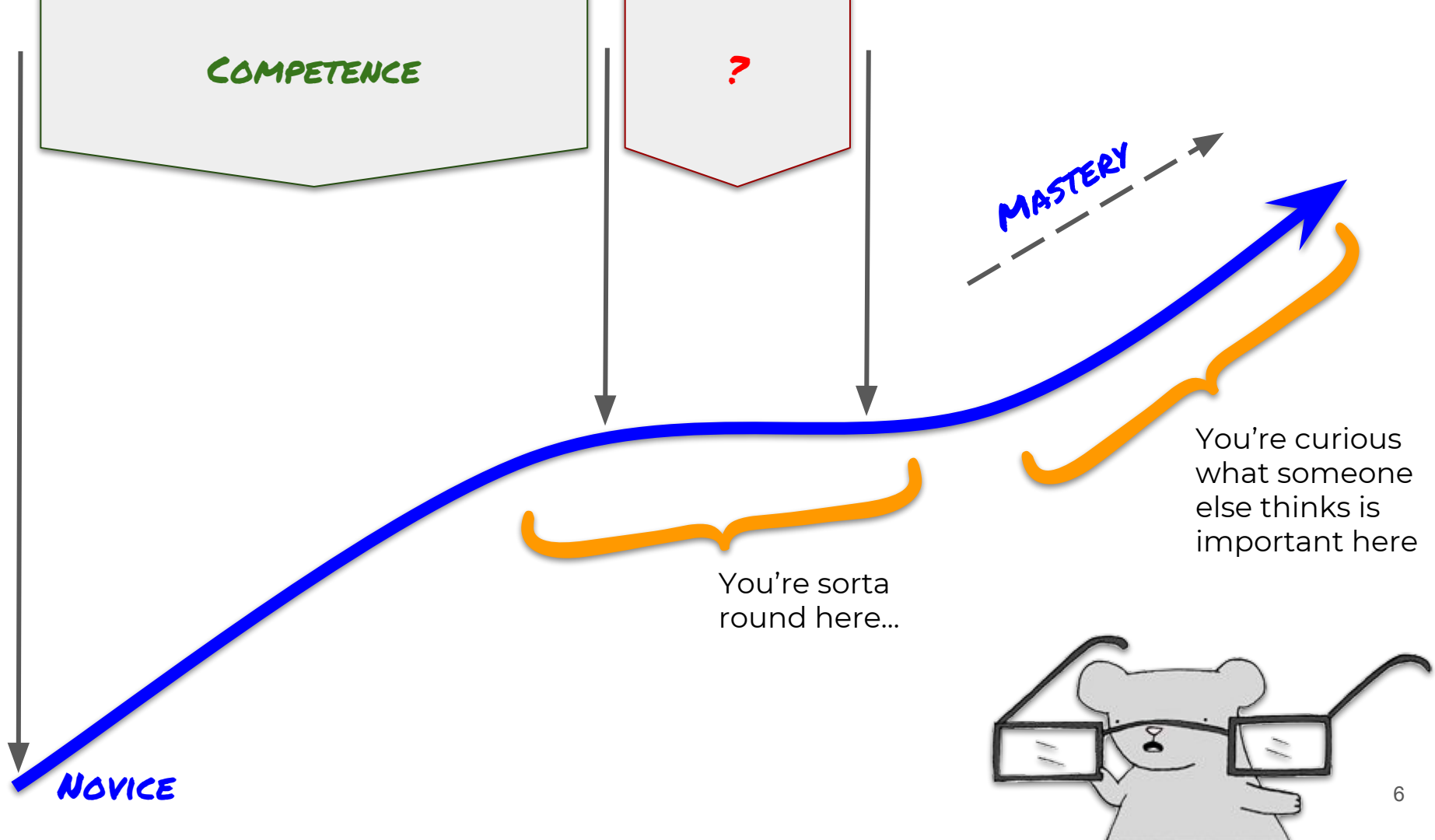
poorlydrawnlines.com



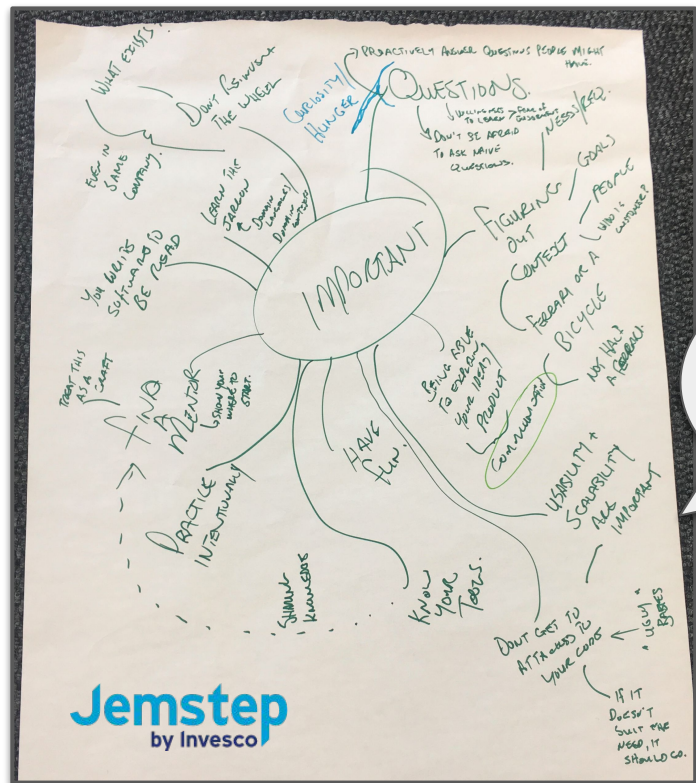
# When creating software, what **Really** Matters?



What stage  
are you at in  
your career?



# COMPETENCE



Figuring out context.

Knowing your **tools**

Not being afraid to ask "naive" question

Not reinventing the wheel

Finding a **mentor**

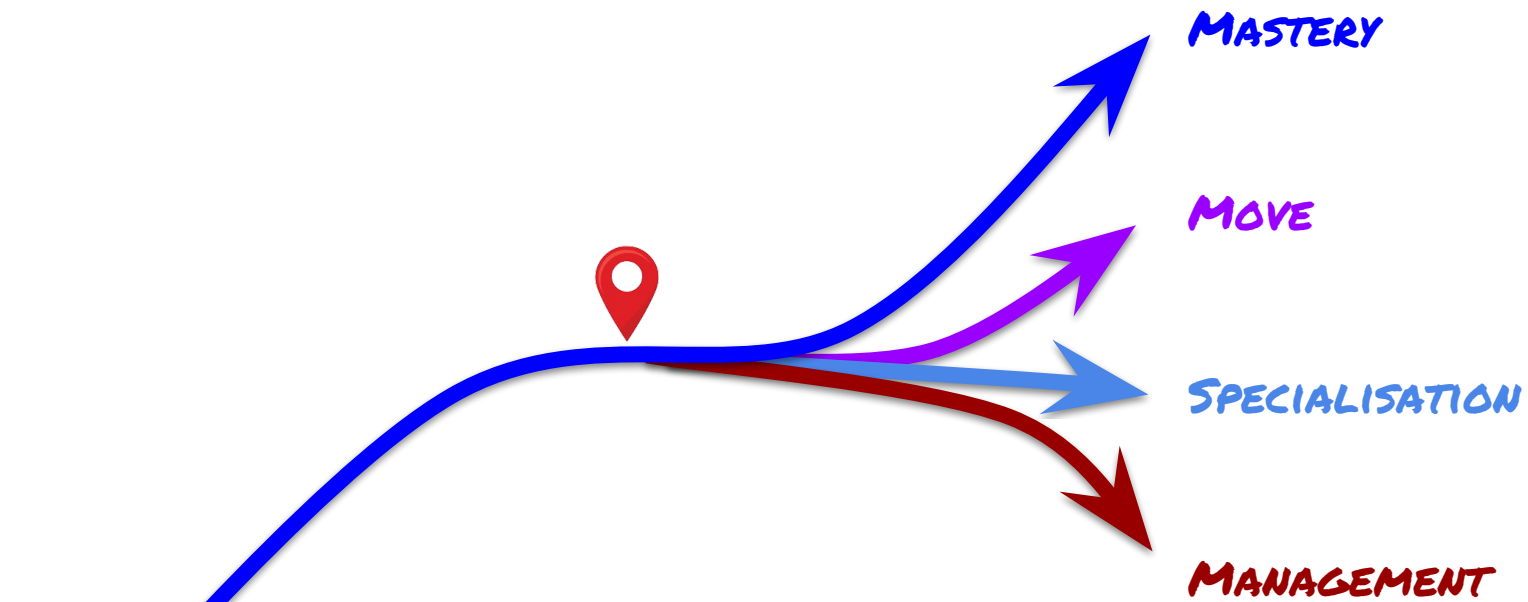
**Practicing** Intentionally

Treating software dev as a **craft**

Having **fun!**

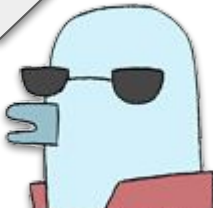
Not getting too attached to your code

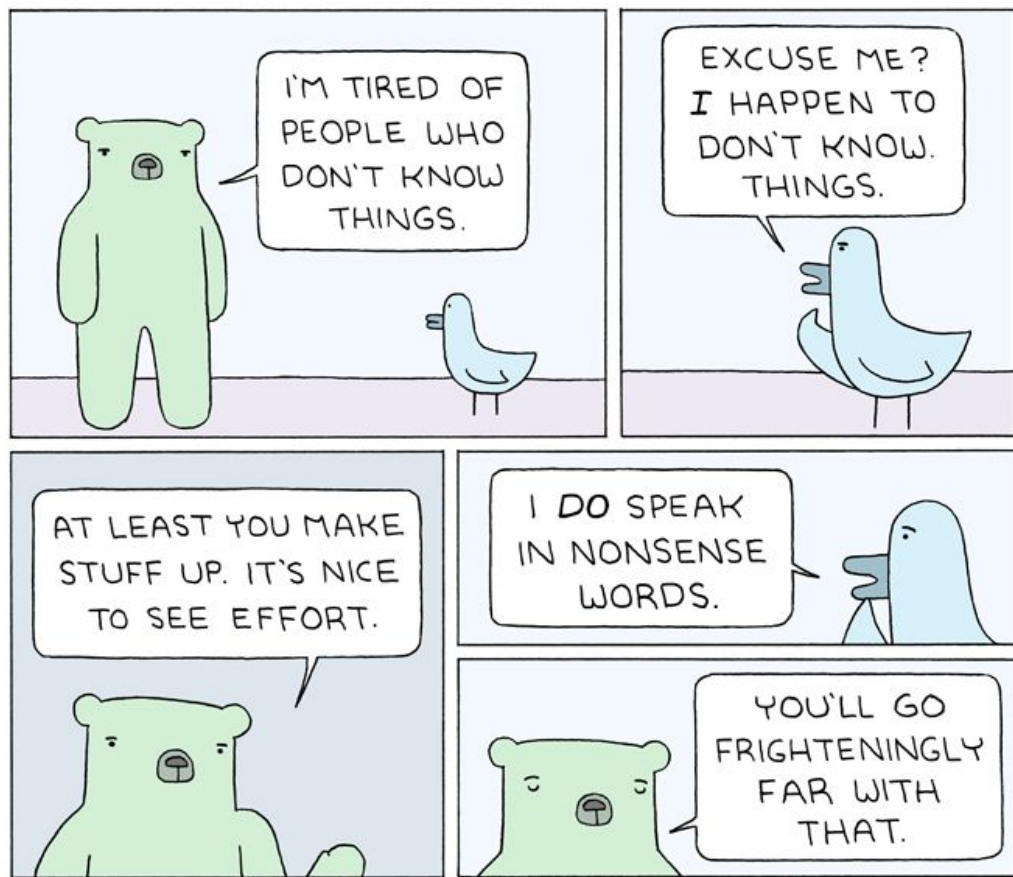
Learning the **jargon**



# Mastery of **What?**

Narrative  
Reality  
Humans  
Systems  
Knowledge  
Flow





poorlydrawnlines.com

# Narrative

How did I come to believe the things I believe?

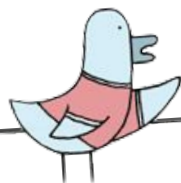
# When creating software, what **Really** Matters?

Awareness of **Narrative** helps me create better software

- Dim American Users
- Why won't they see I'm here to help?
- We do TDD.



Responsibility for client growth ↑	<b>Counsellor</b> "You do it. I will be your sounding board"	<b>Coach</b> "You did well, you can add this next time."	<b>Partner</b> "We will do it together and learn from each other."
	<b>Facilitator</b> "You do it, I will attend to the process."	<b>Teacher</b> "Here are some principles you can use to solve problems of this type."	<b>Modeller</b> "I will do it; you watch so you can learn from me."
	<b>Reflective Observer</b> "You do it; I will watch and tell you what I see and hear."	<b>Technical Advisor</b> "I will answer your questions as you go along."	<b>Hands-on Expert</b> "I will do it for you. I will tell you what to do."
	→ Responsibility for client results		





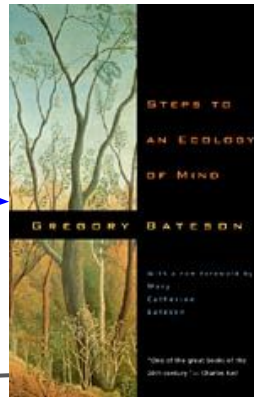
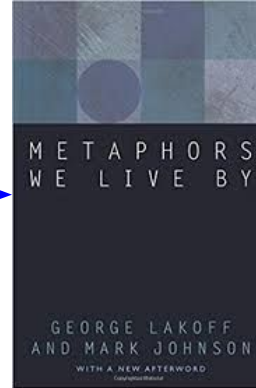
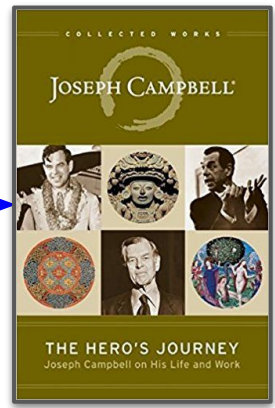
Joseph Campbell



George Lakoff



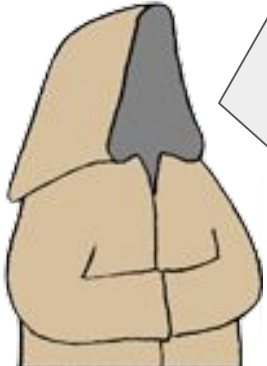
Gregory Bateson



# When creating software, what **Really** Matters?

I learn from the ***people***,  
not the label.

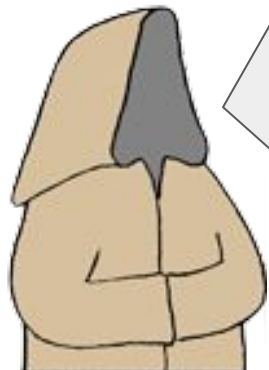
Pro Tip™

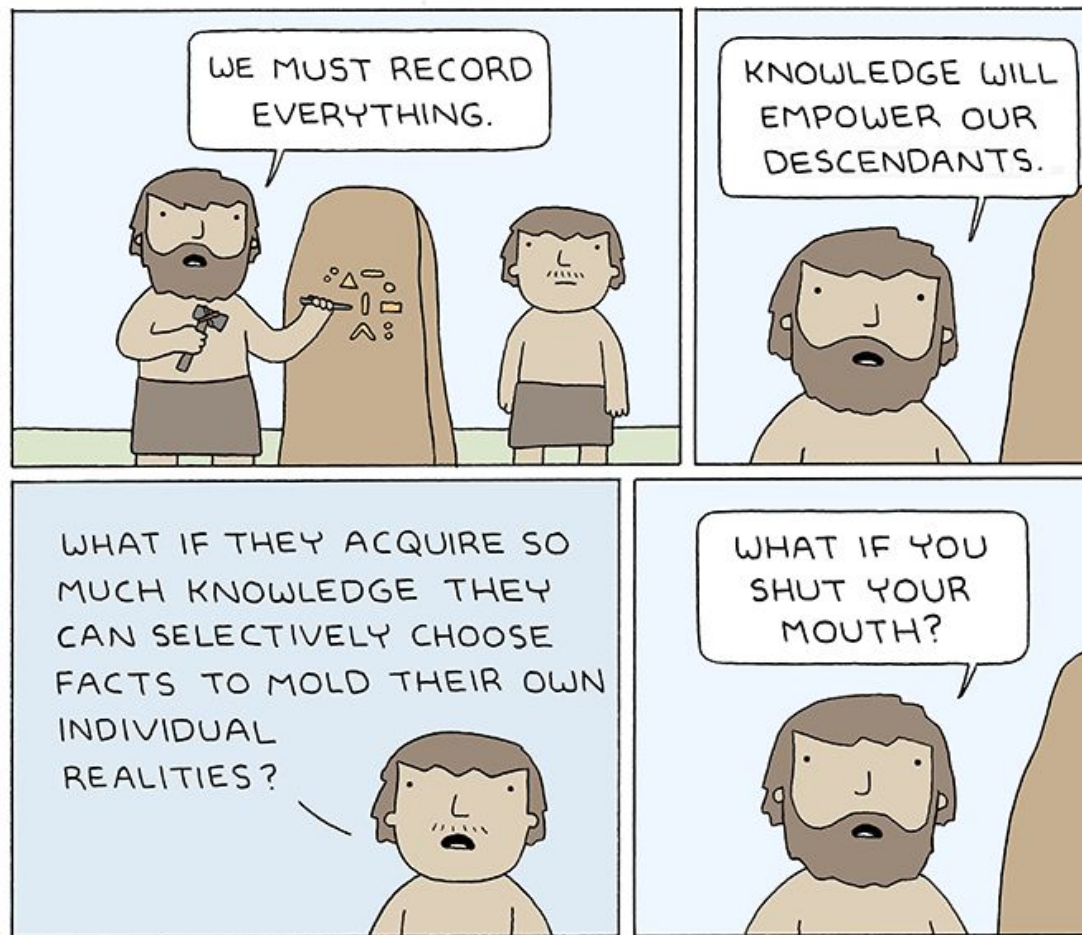


# When creating software, what **Really** Matters?

I read to ***improve my compiler,***  
not to memorize facts.

Pro Tip™





poorlydrawnlines.com

# Narrative Reality

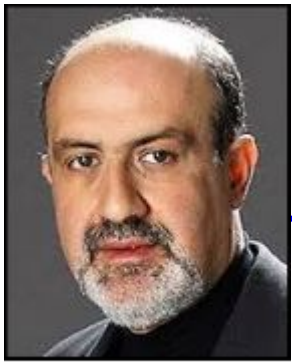
How do I know the difference  
between what I believe,  
and reality?

# When creating software, what **Really** Matters?

Awareness of **Reality** helps me create better software

- Retros at ABC Bank
- Project Managers at XYZ Bank
- Single Backlog at Jemstep

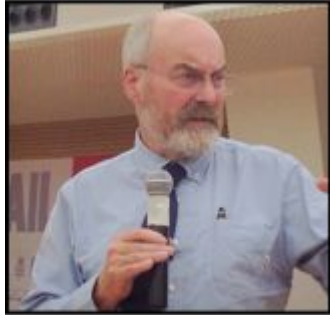




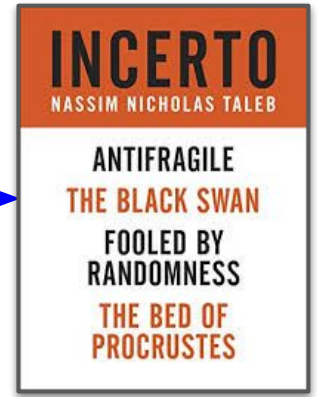
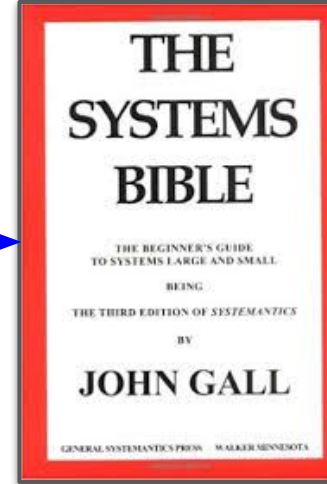
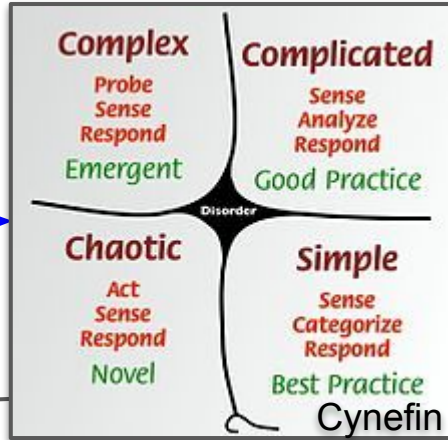
Nassim Taleb



John Gall



Dave Snowden

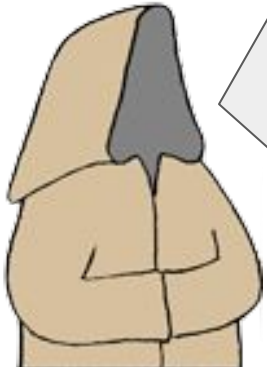


# When creating software, what **Really** Matters?

“I respect facts,  
but impressions are  
what I live with”

- John Gall

Pro Tip™

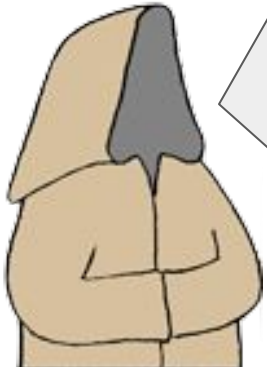


# When creating software, what **Really** Matters?

“It is in the doing of the work that we discover the work that we must do.  
Doing exposes reality.”

- Woody Zuill

Pro Tip™



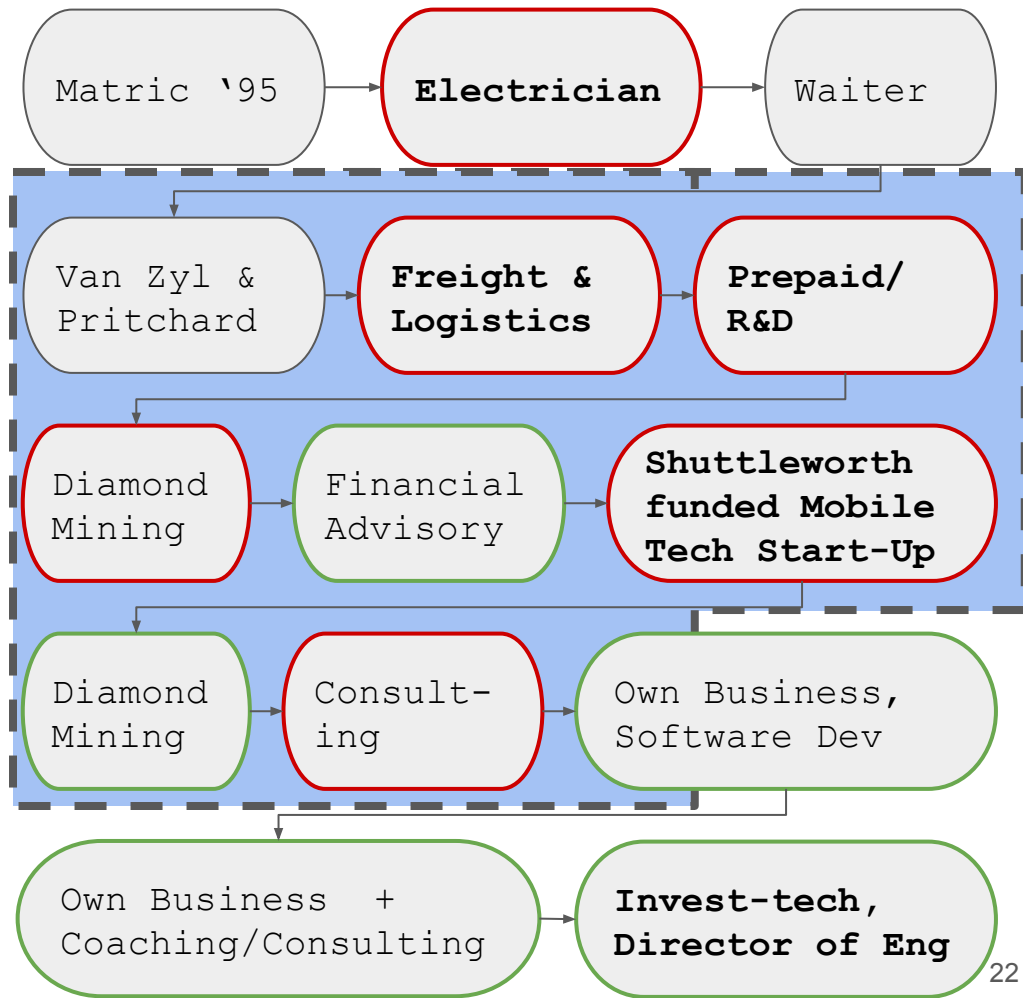
## BASIC LIFE TUTORIAL



poorlydrawnlines.com

# Narrative Reality Humans

Who are all these people  
around me?



# When creating software, what **Really** Matters?

## Awareness of **Humans** helps me create better software

- Inflicting helping on team BL vs improving team KB
- WOO
- Project Managers at XYZ Bank + NVC





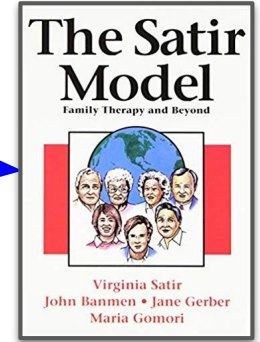
Esther Derby



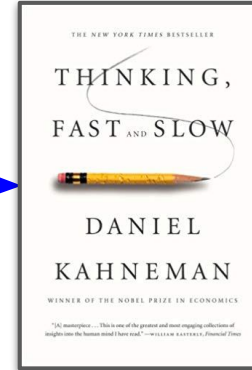
Gerald M Weinberg



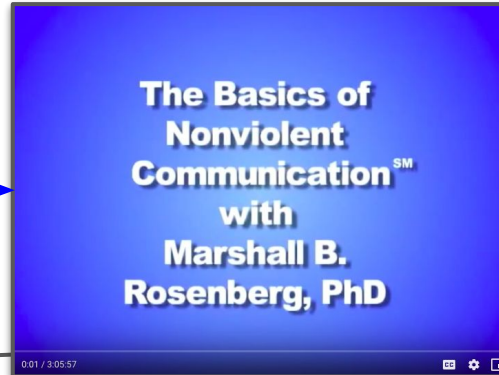
Virginia Satir



Daniel Kahneman



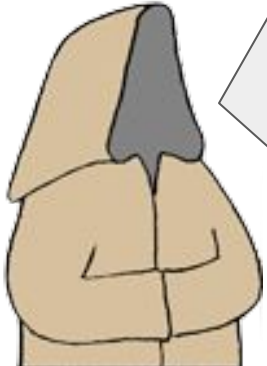
Marshall Rosenberg



# When creating software, what **Really** Matters?

Everybody is doing their best  
with what they have.

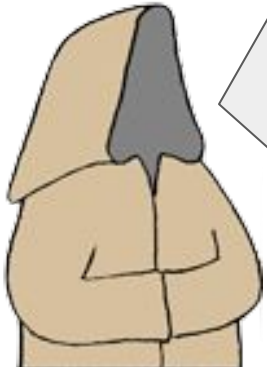
Pro Tip™

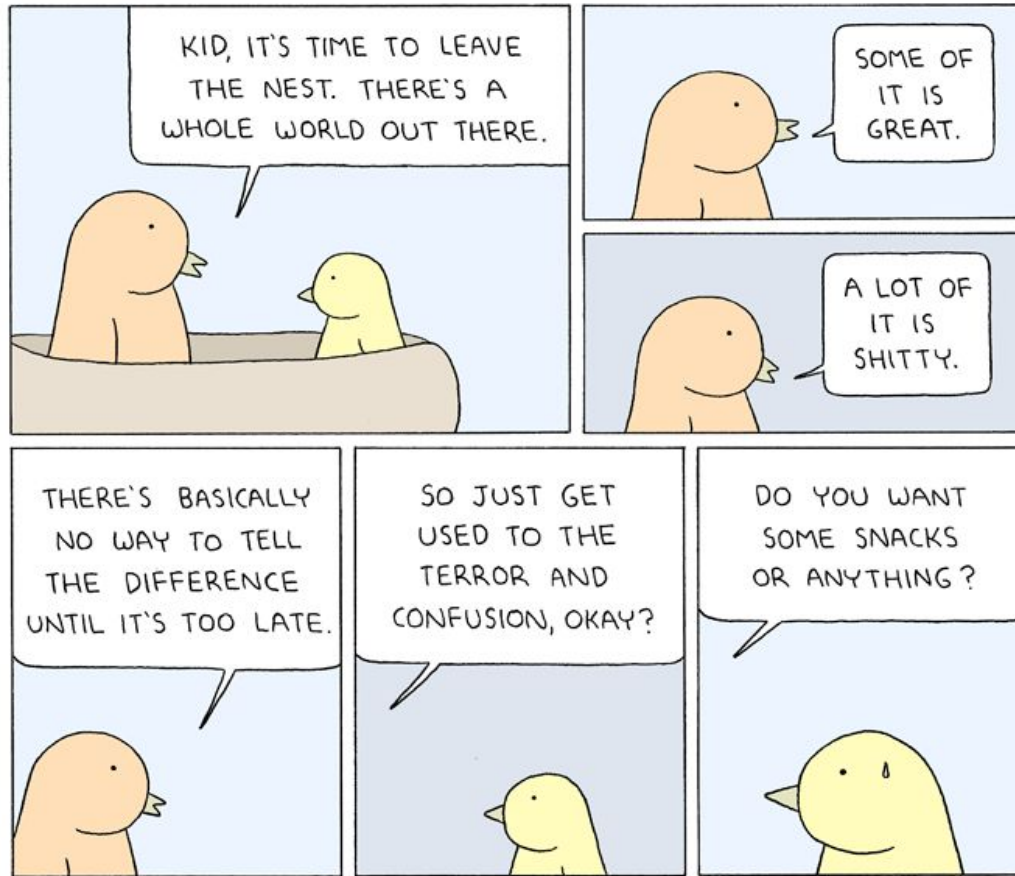


# When creating software, what **Really** Matters?

**Growing my (human) network** has been hugely rewarding - to me and those within my network.

Pro Tip™





poorlydrawnlines.com

# Narrative Reality Humans Systems

How come these people are so smart, but what is happening so not?

# When creating software, what **Really** Matters?

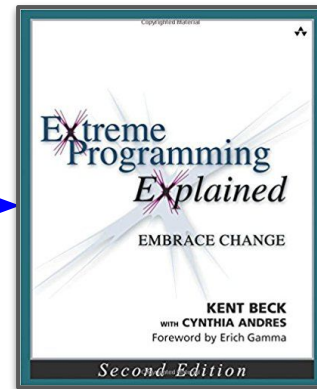
Awareness of **Systems** helps me  
create better ~~software~~

- XL Program - Explain a Pen
- Infrastructure team at XYZ Bank
- Spine Model

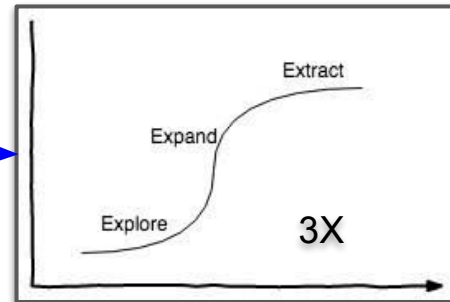




Kent Beck



Glenda Eoyang



Russell Ackoff

Complex block containing logos for Pegasus Classics, Systems Thinker, and a title 'FROM MECHANISTIC TO SOCIAL SYSTEMIC THINKING'.

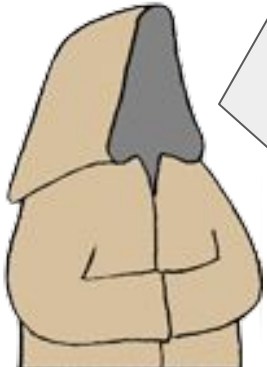
Complex block containing a title 'HUMAN SYSTEMS DYNAMICS INSTITUTE', a subtitle 'Caught in the Middle', and a large 'CDE' logo.



# When creating software, what **Really** Matters?

Until I gain enough experience in a particular domain, practices and tools are what I can see.

Pro Tip™





poorlydrawnlines.com

# Narrative Reality Humans Systems **Knowledge**

Why do we suck at documenting, storing and transferring knowledge?

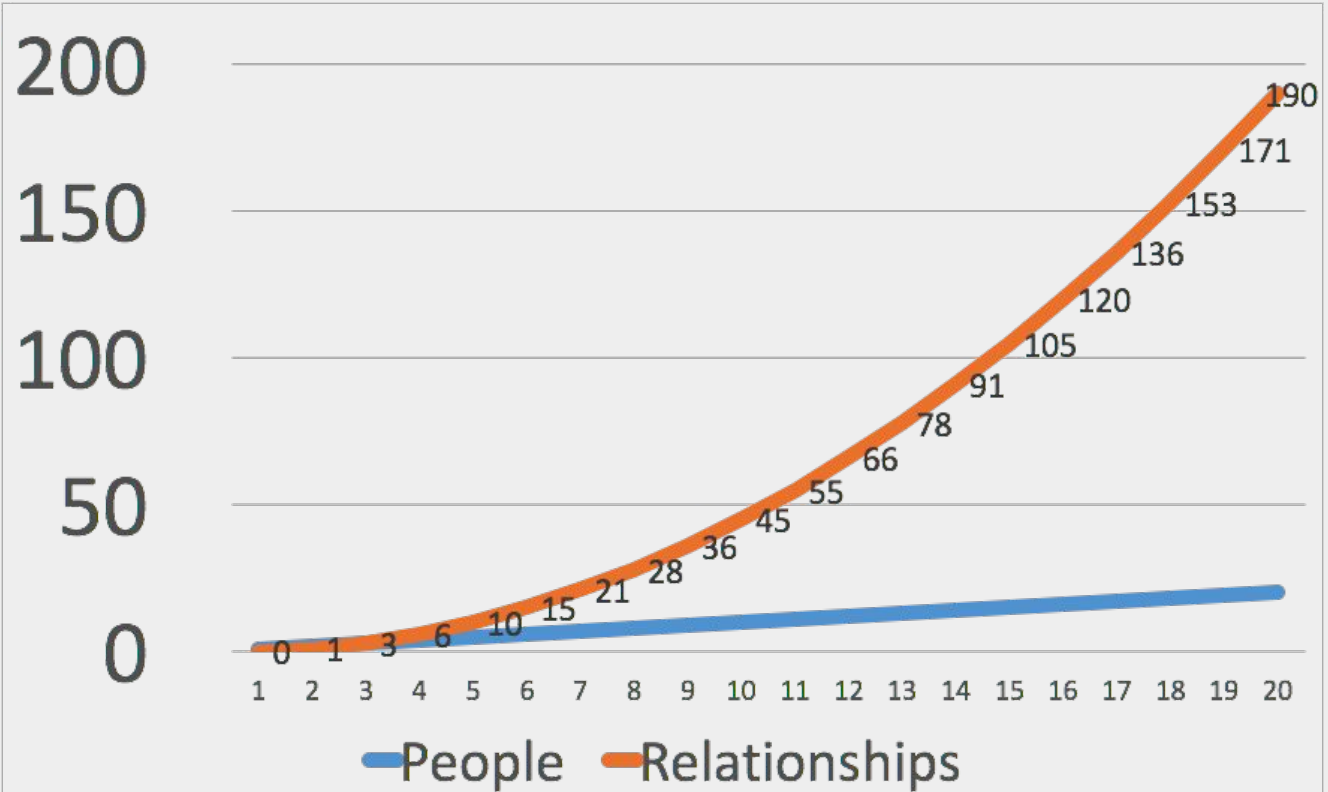
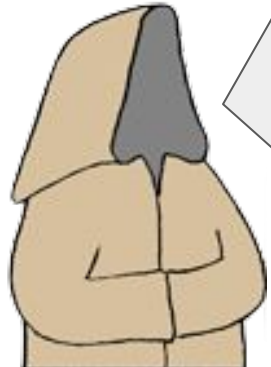
# When creating software, what **Really** Matters?

Awareness of **Knowledge** helps me create better software

- “We’ve gotten really good at handing over”
- Mob programming in Jemstep
- Performance management at Jemstep



# When creating software, what **Really** Matters?





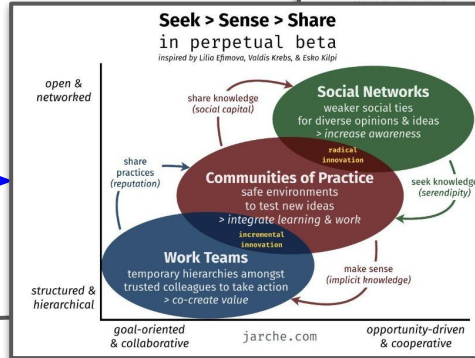
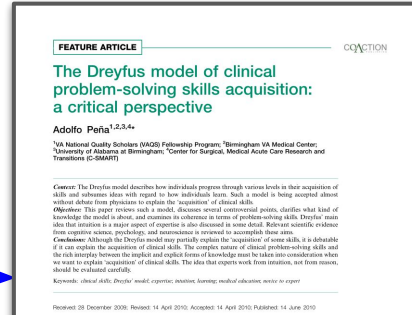
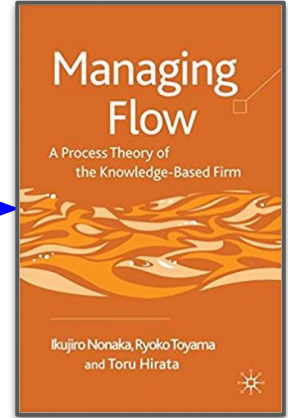
Ikujiro Nonaka



Hubert Dreyfus



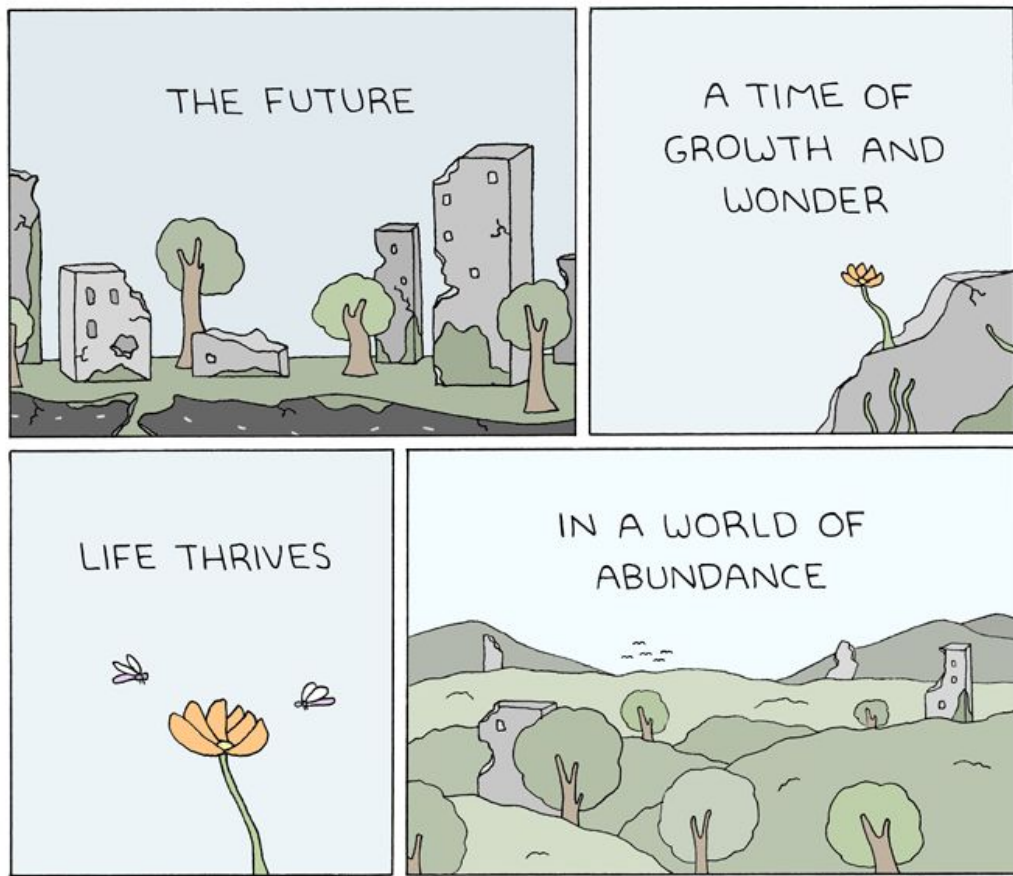
Harold Jarche



imitation in decision making and develop their own rules to formalize plans. Expertise is characterized by a fluid performance that happens unconsciously, automatically, and no longer depends on explicit knowledge. Thus, the progression is envisaged as a gradual transition from a rigid adherence to taught rules and procedures through a largely intuitive mode of operation that relies heavily on deep, implicit knowledge but accepts that sometimes at expert level analytical approaches are still likely to be used when an intuitive approach fails initially.

This model, a product of philosophical deliberation and phenomenological research, was initially adapted by Benne and other nursing educators to explain the development of nursing skills (1). However, this was not without debate, which still remains. Hargrove and Lane criticized Benne's model, a linear model of skill acquisition that cannot sufficiently explain the everyday experiences of learning (2). Thompson (3), Parke (4), and Ridge (5) criticized Benne's and Dreyfus' models because of their apparent absence of social structure or social knowledge. English pointed out that Benne's and





poorlydrawnlines.com

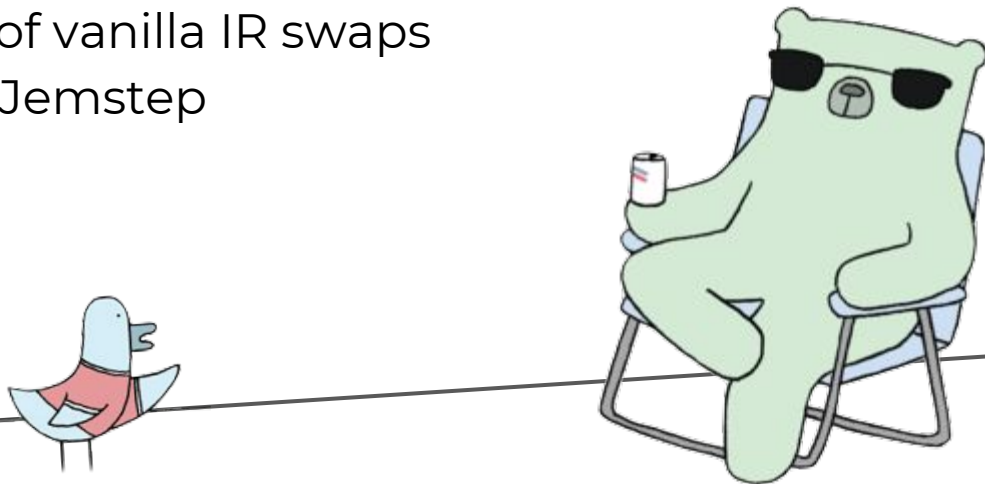
# Narrative Reality Humans Systems Knowledge **Flow**

With so much working happening, why does so little get done?

# When creating software, what **Really** Matters?

Awareness of **Flow** helps me create better software

- Quants not happy
- Pull vs Push queuing of vanilla IR swaps
- Testing bottleneck at Jemstep

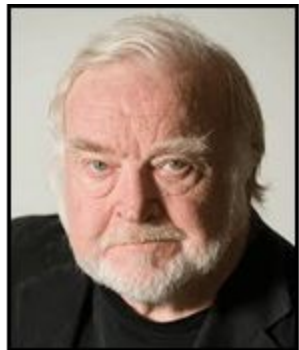




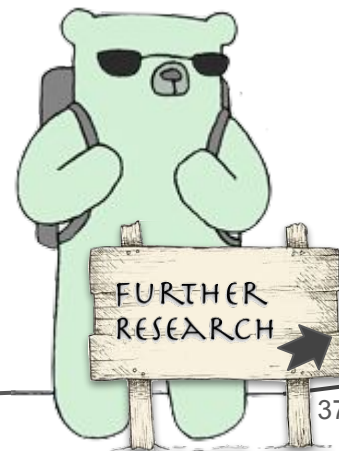
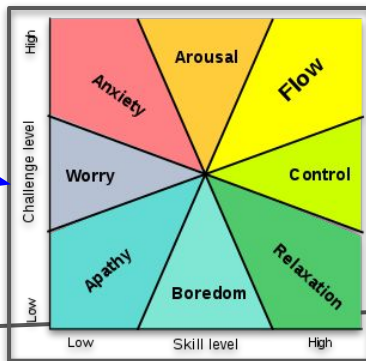
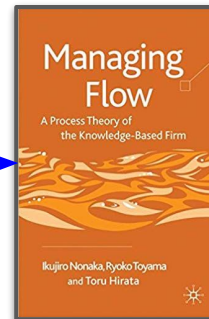
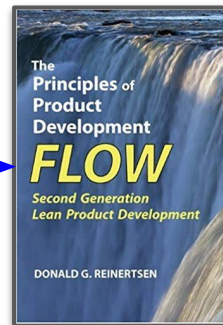
Don Reinertsen



Ikujiro Nonaka



Mihaly Csikszentmihalyi



When creating software, what **Really** Matters?



# Discussion Points?

Narrative  
Reality  
Humans  
Systems  
Knowledge  
Flow



@kevintrethewey



kevint@gmail.com  
kevint@jemstep.com



# When creating software, what **Really** Matters?

Iterate

Do it. Reflect. Improve. Do it again.

Pro Tip™

