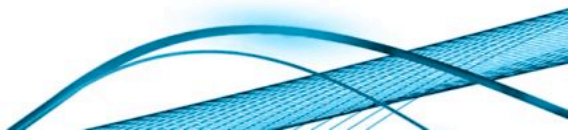
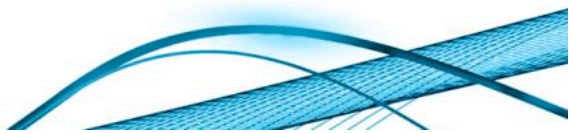
Building the Next Generation of Technical Leaders

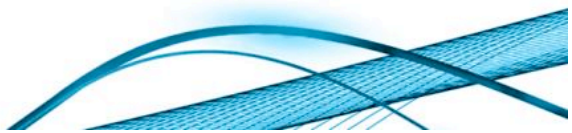
patrick.kua@thoughtworks.com
http://www.thekua.com/atwork
Twitter: @patkua

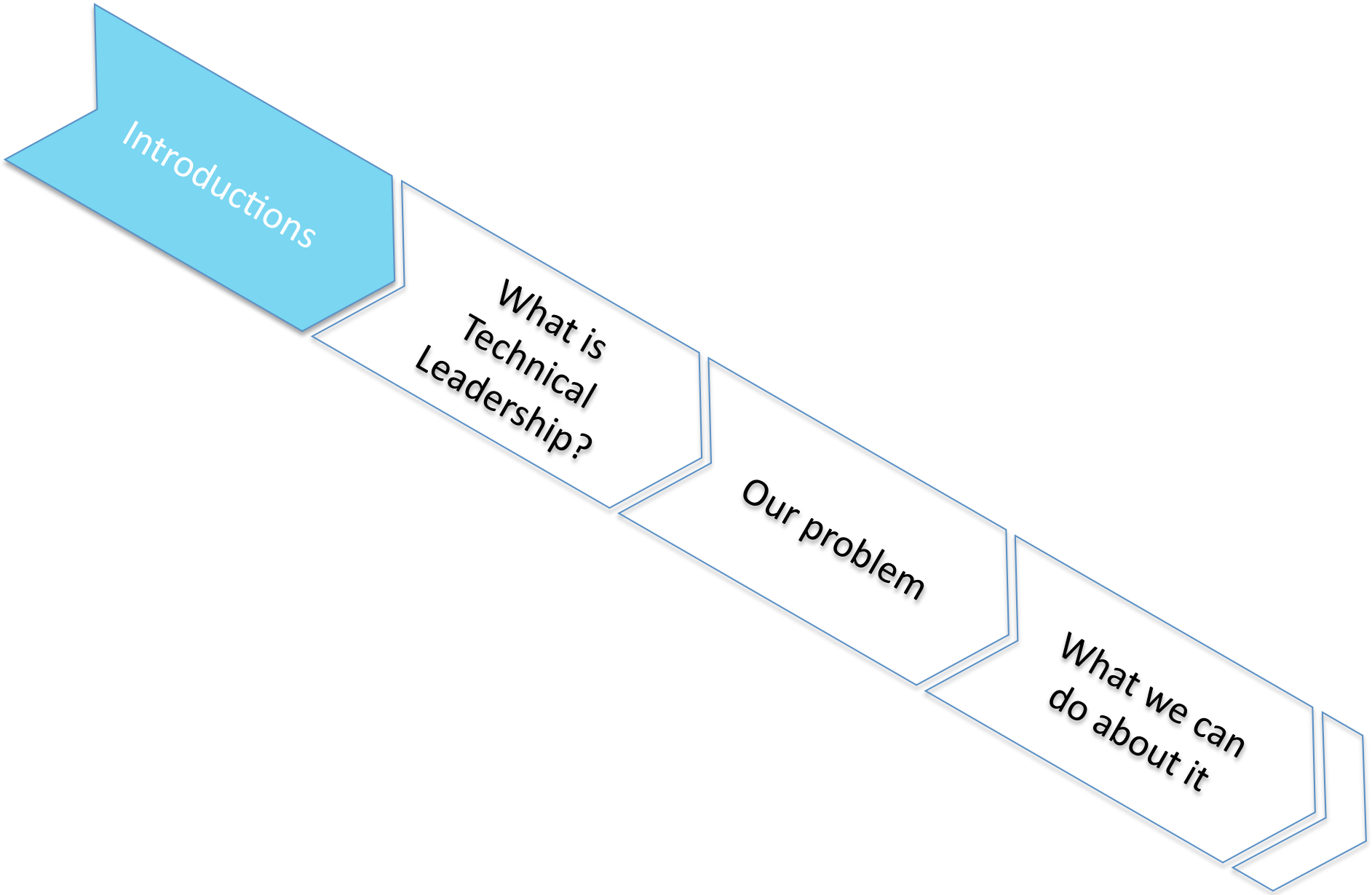


Housekeeping



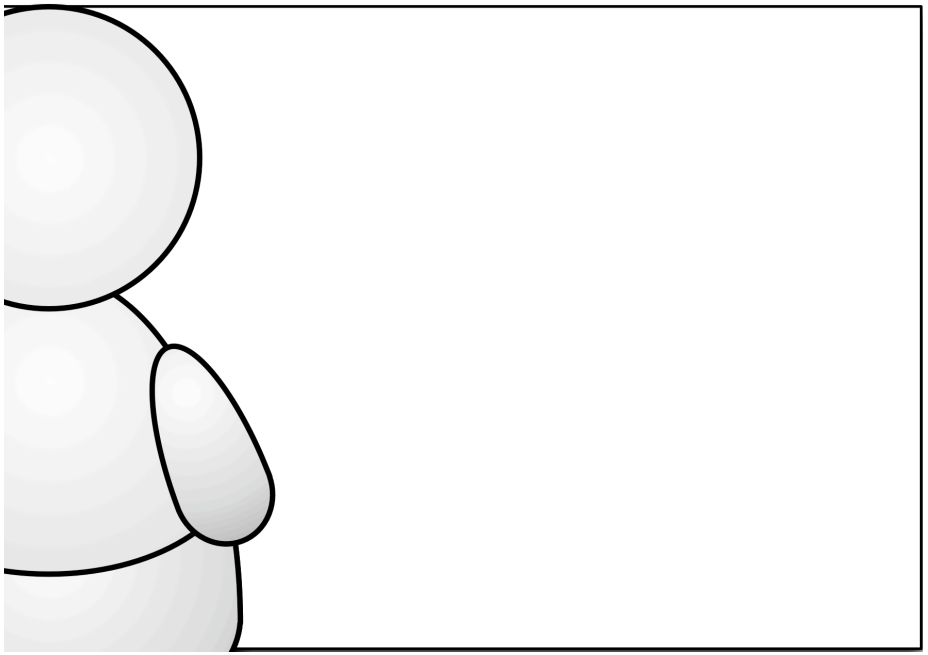
Our Journey



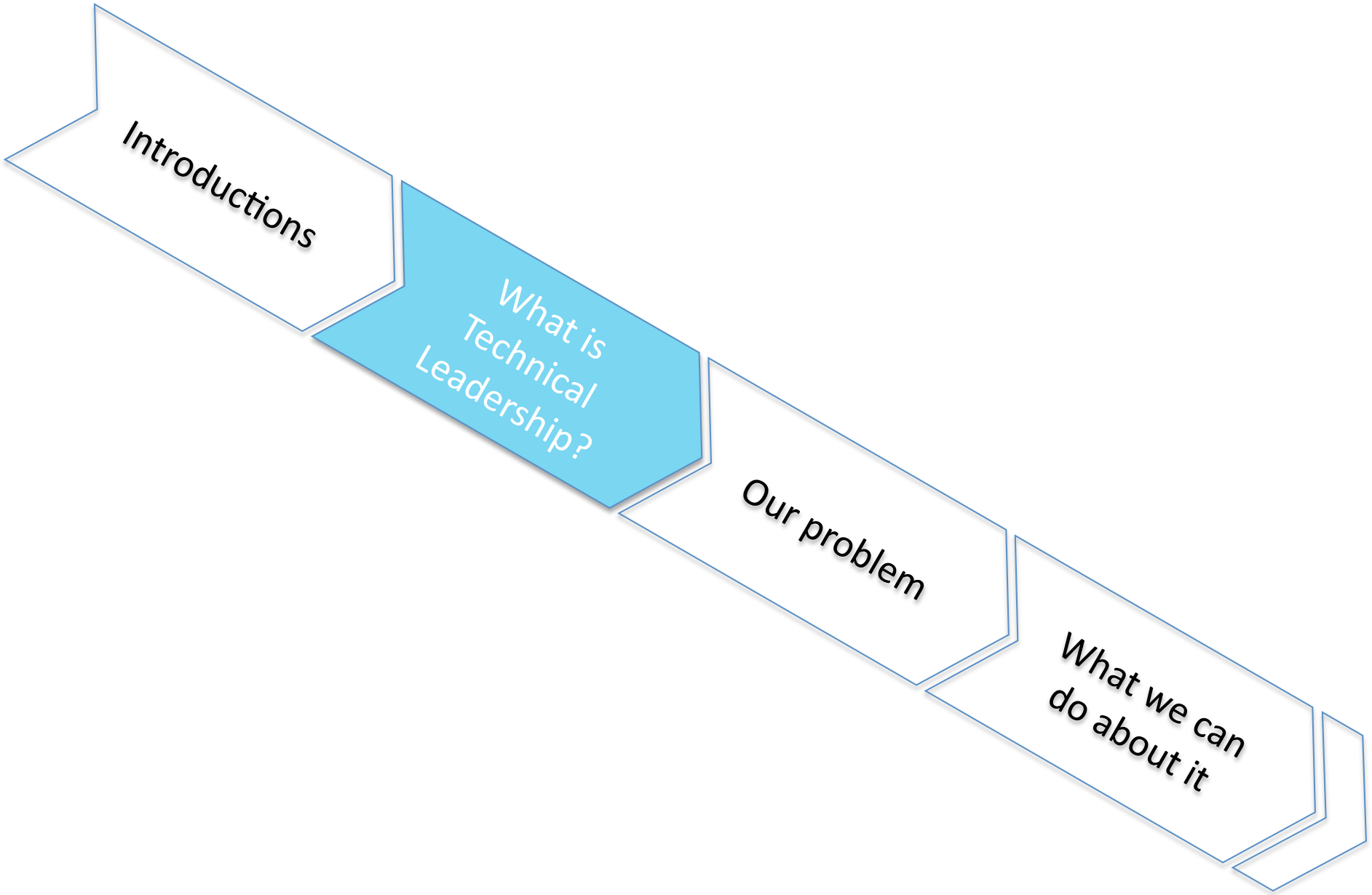


ThoughtWorks®

Who am I?

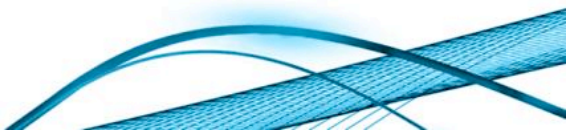


Who are you?



ThoughtWorks®

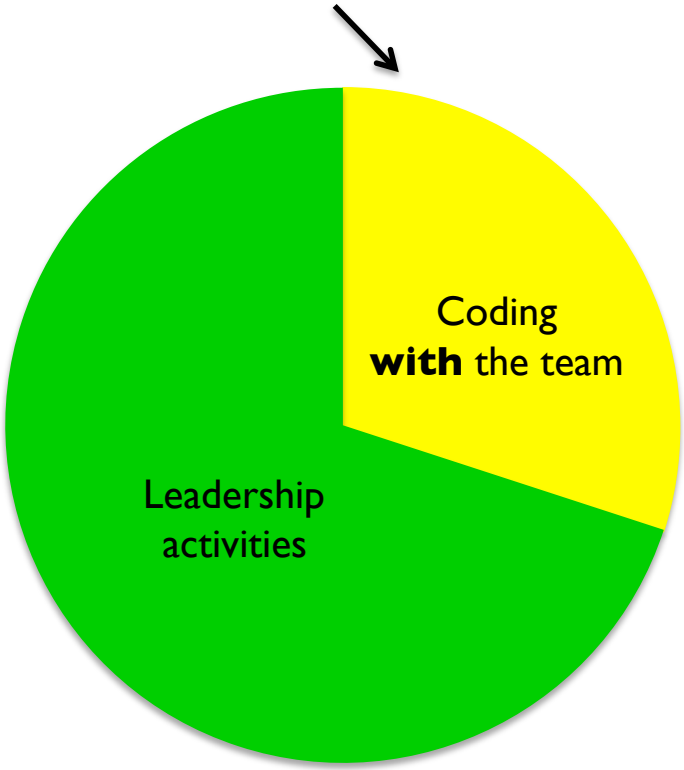
Ubiquitous Language!



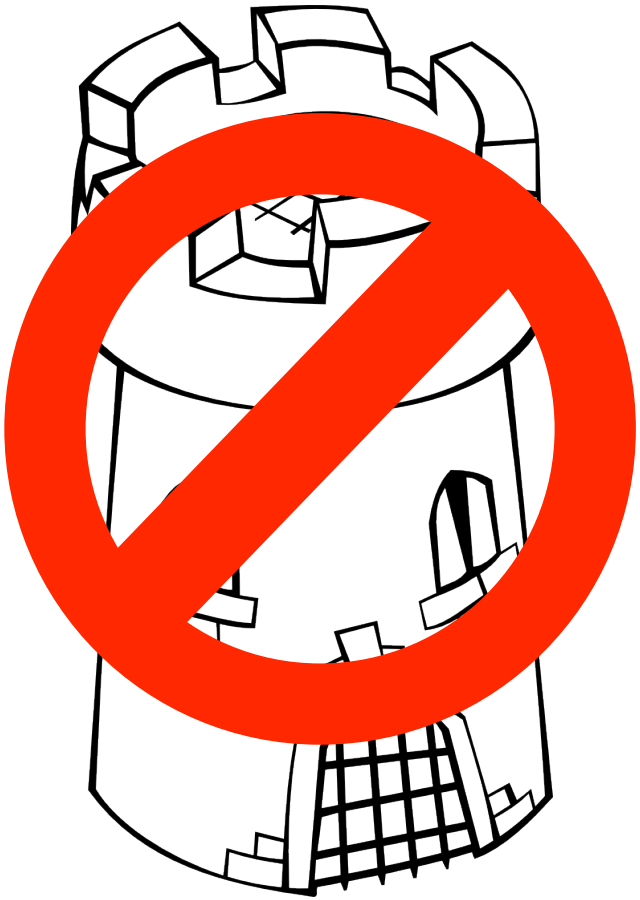
Define…

Technical Leader

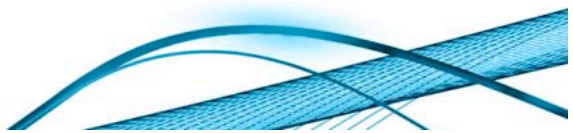
A leader who spends at least 30% of their time coding with the team

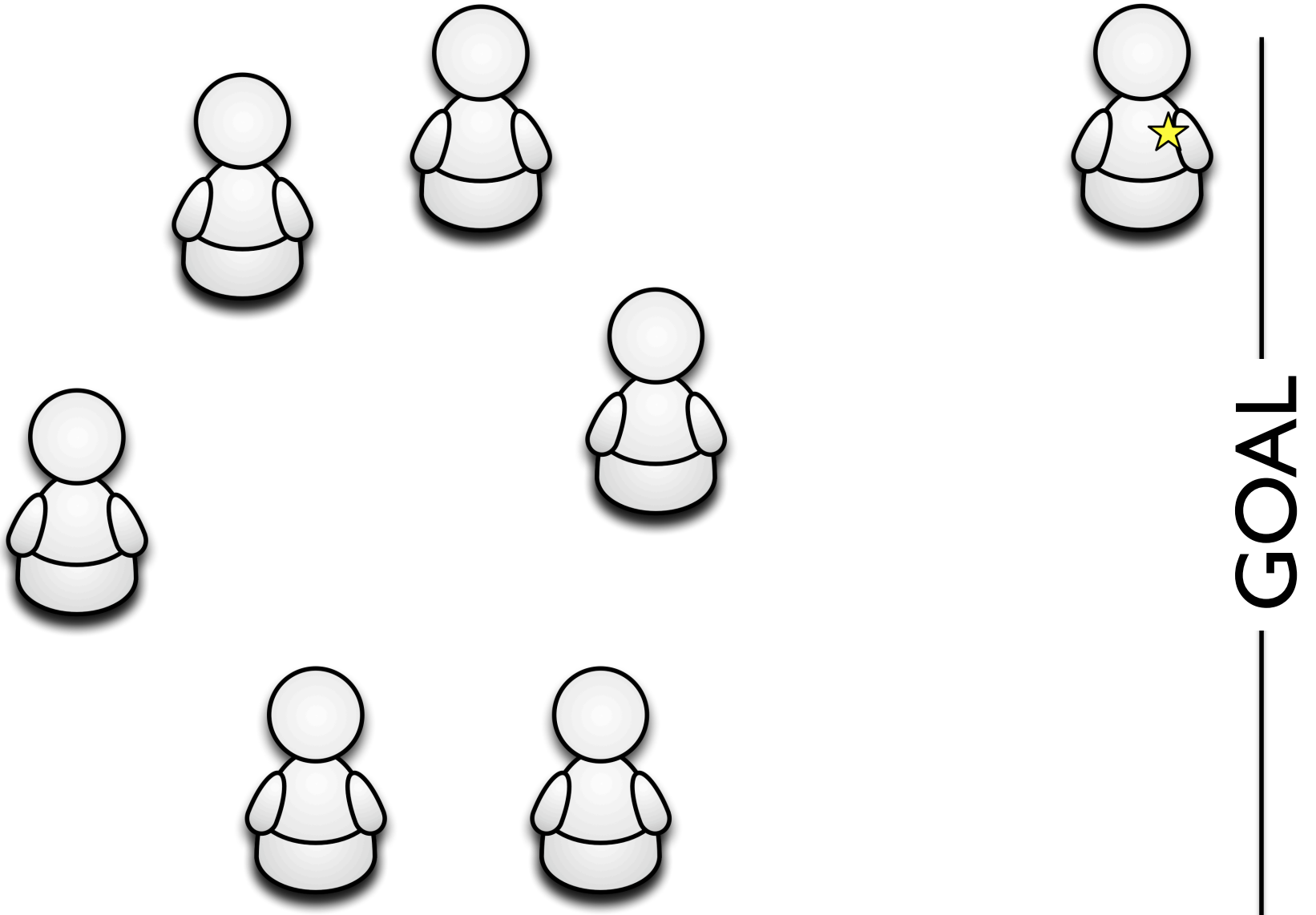


(our definition for today)



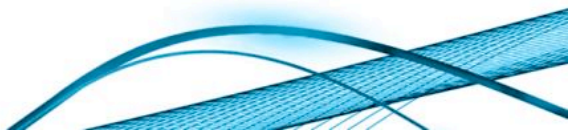
Why do we have Technical Leaders?



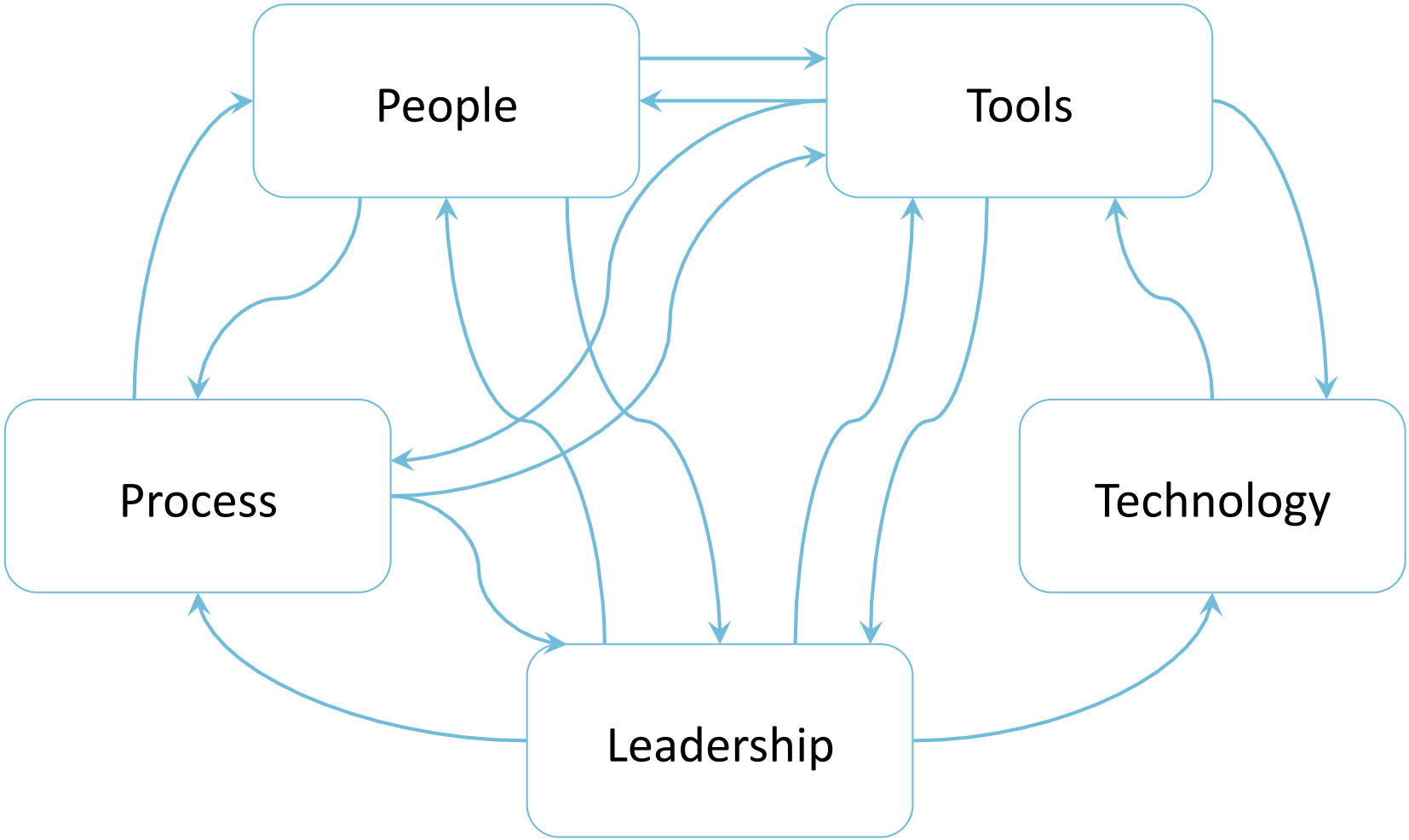


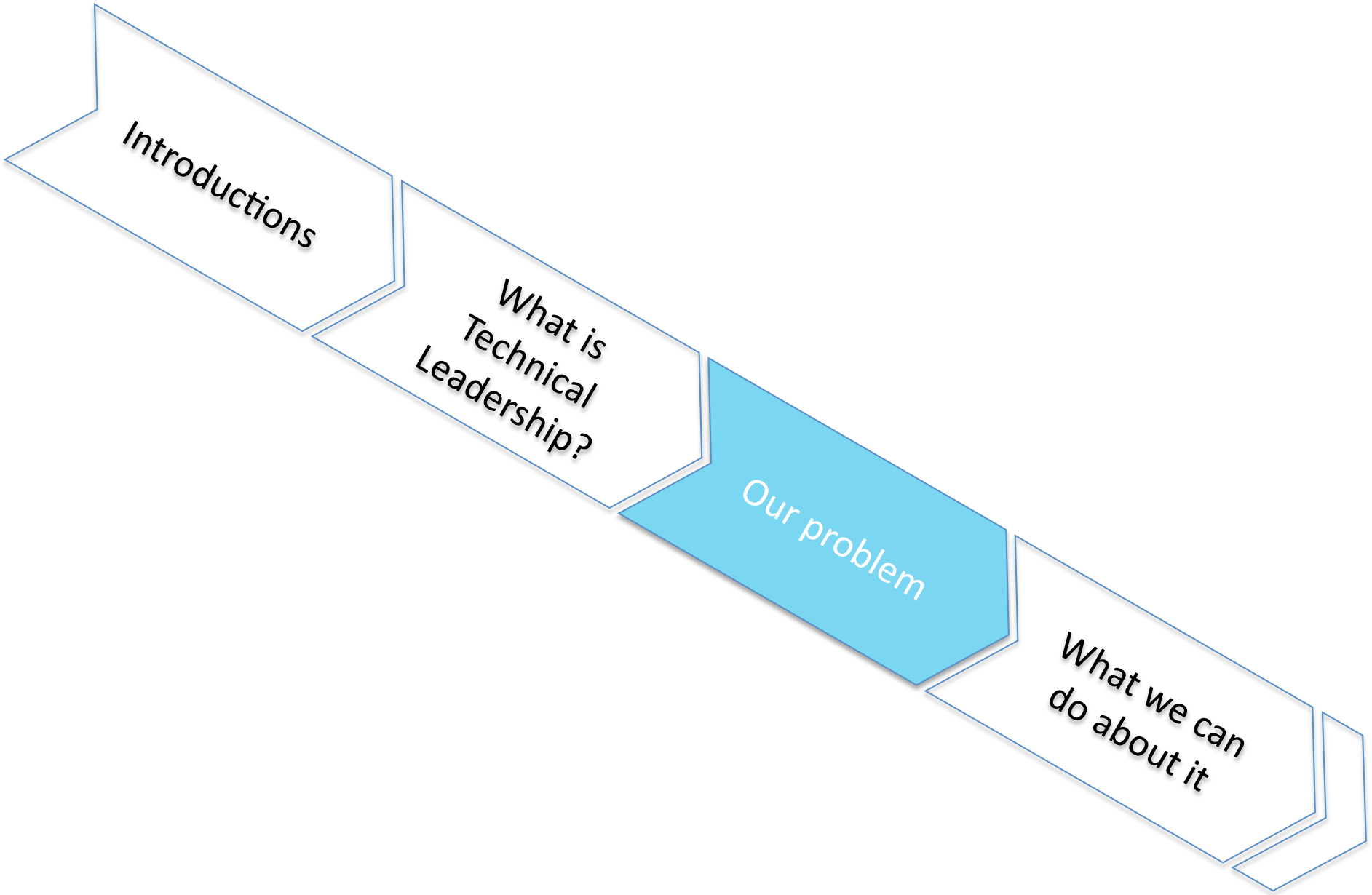
"The unspoken truth about managing geeks"

Source: By Jeff Ello via Computer World - http://bit.ly/15Rm4z

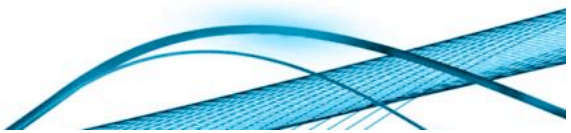


Software systems

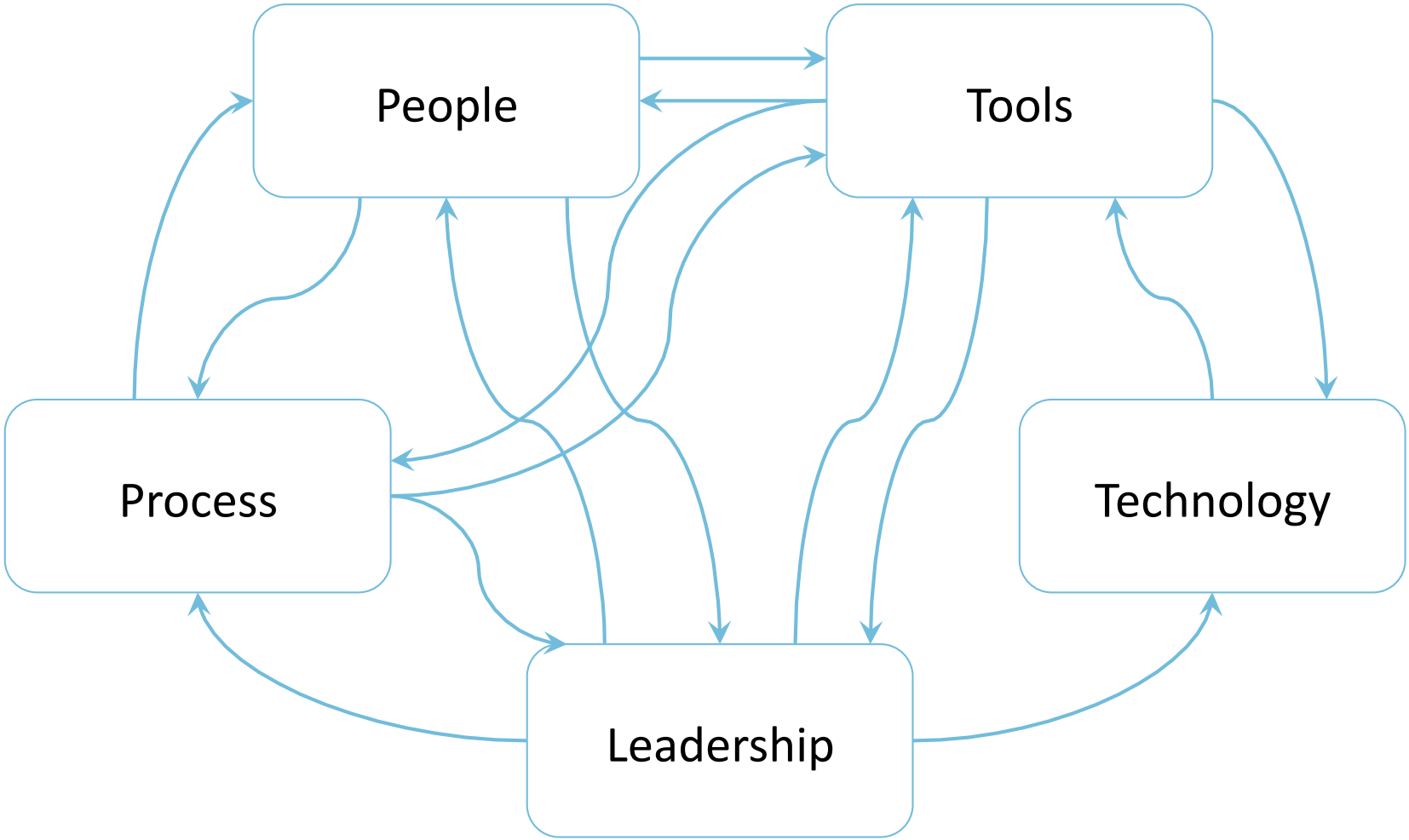



Introductions
What is Technical Leadership?
Our problem
What we can do about it

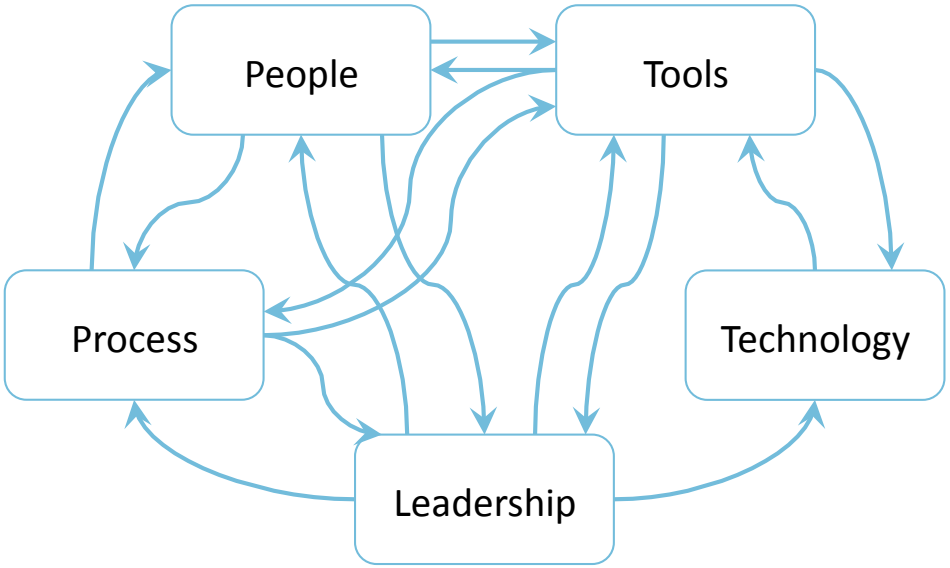
Why do IT projects fail?



Software systems

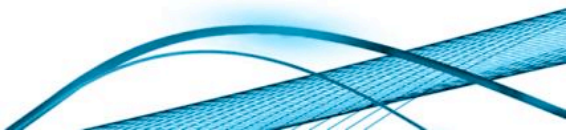


"A bad system will beat a good person every time"



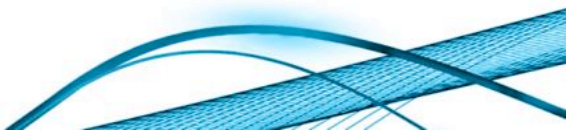
W. Edwards Deming

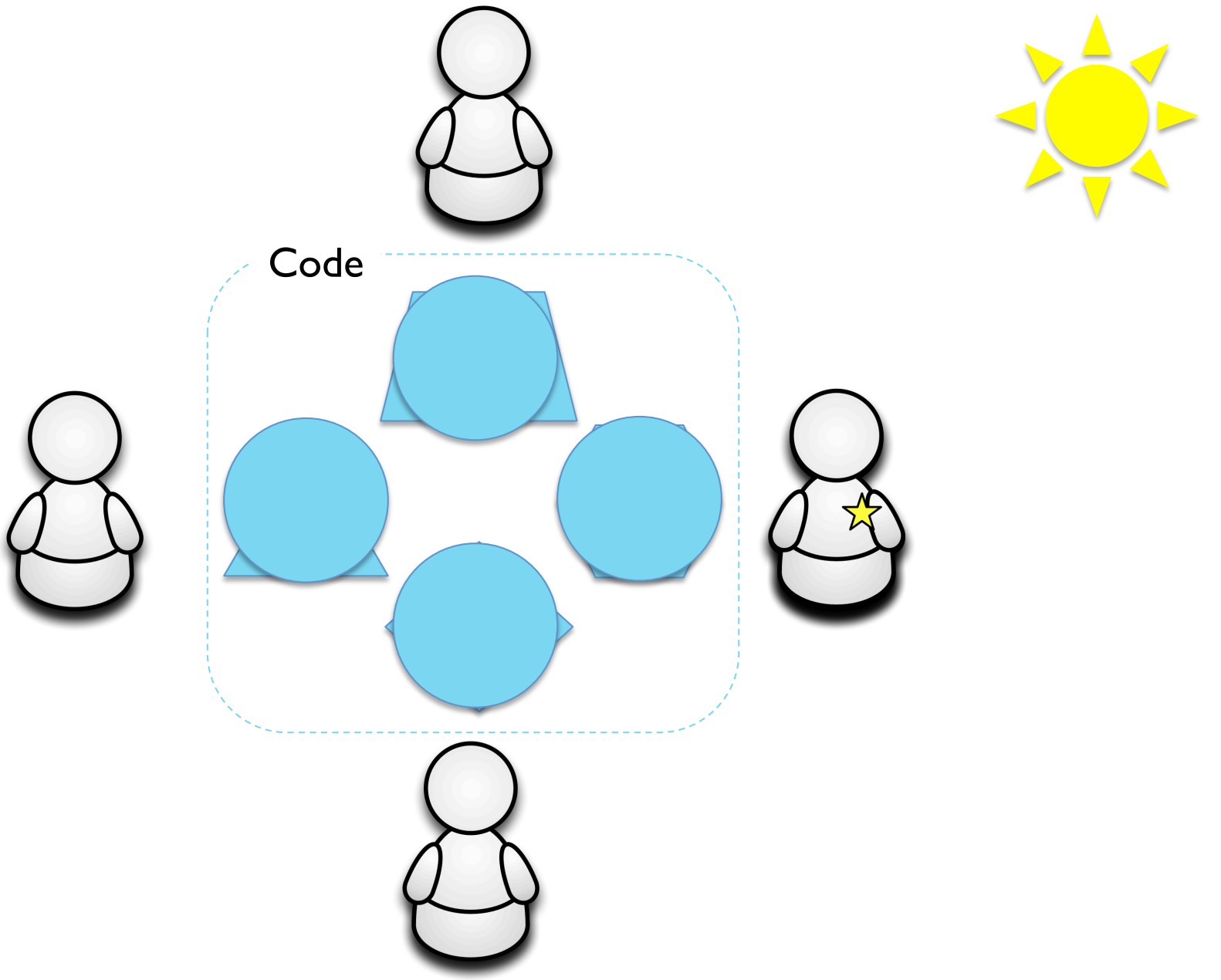
Examples of damaging behaviour...

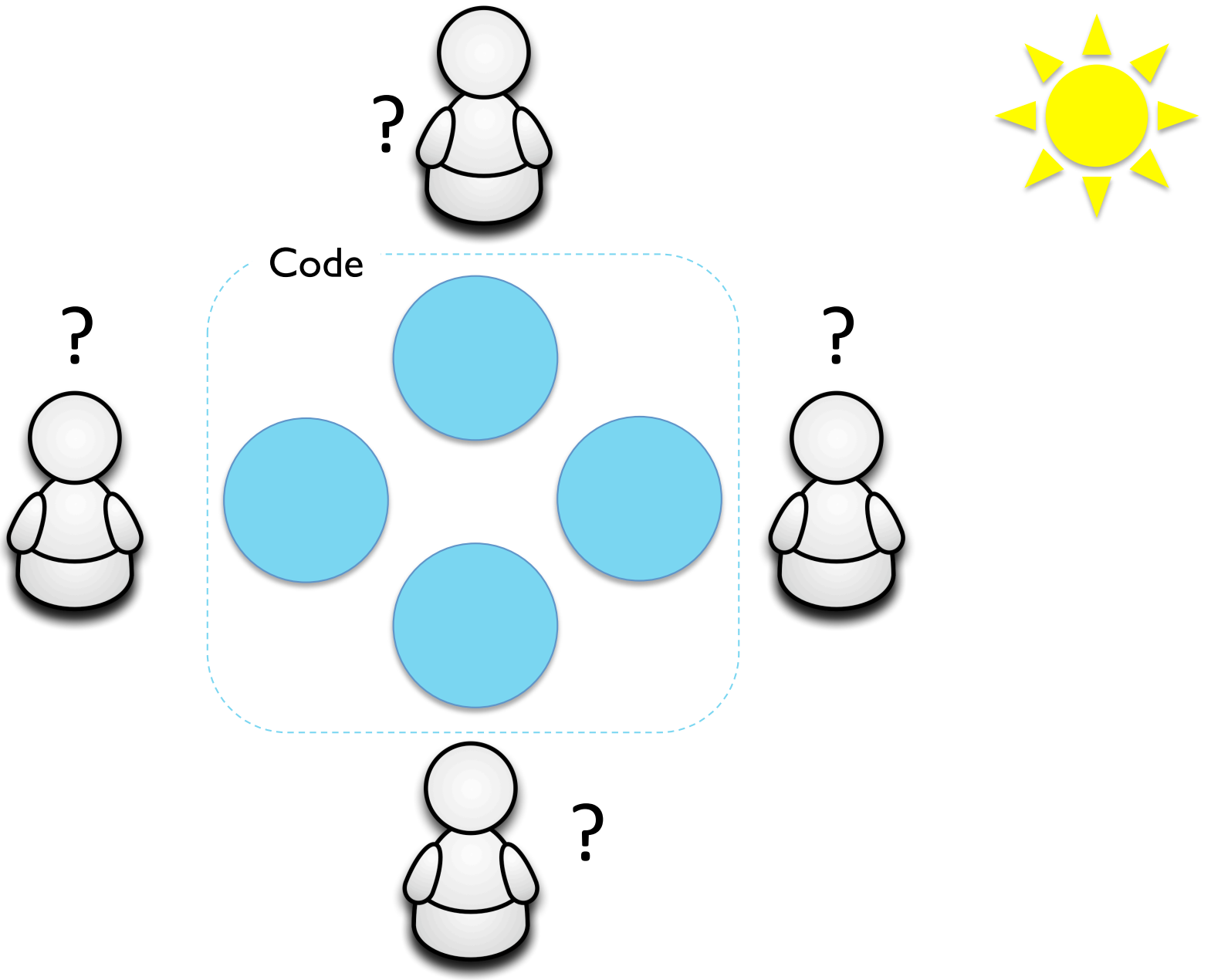


Early morning refactor

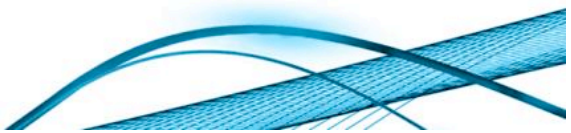
(late evening)



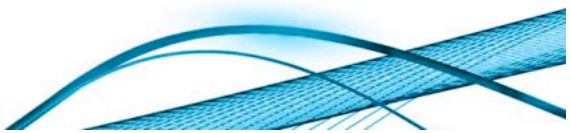




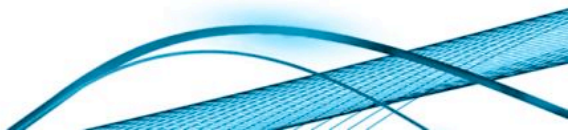
“~~Benevolent~~” Dictator



DIY

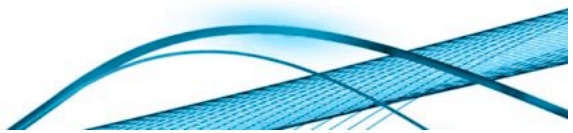


“The hard problems are mine”

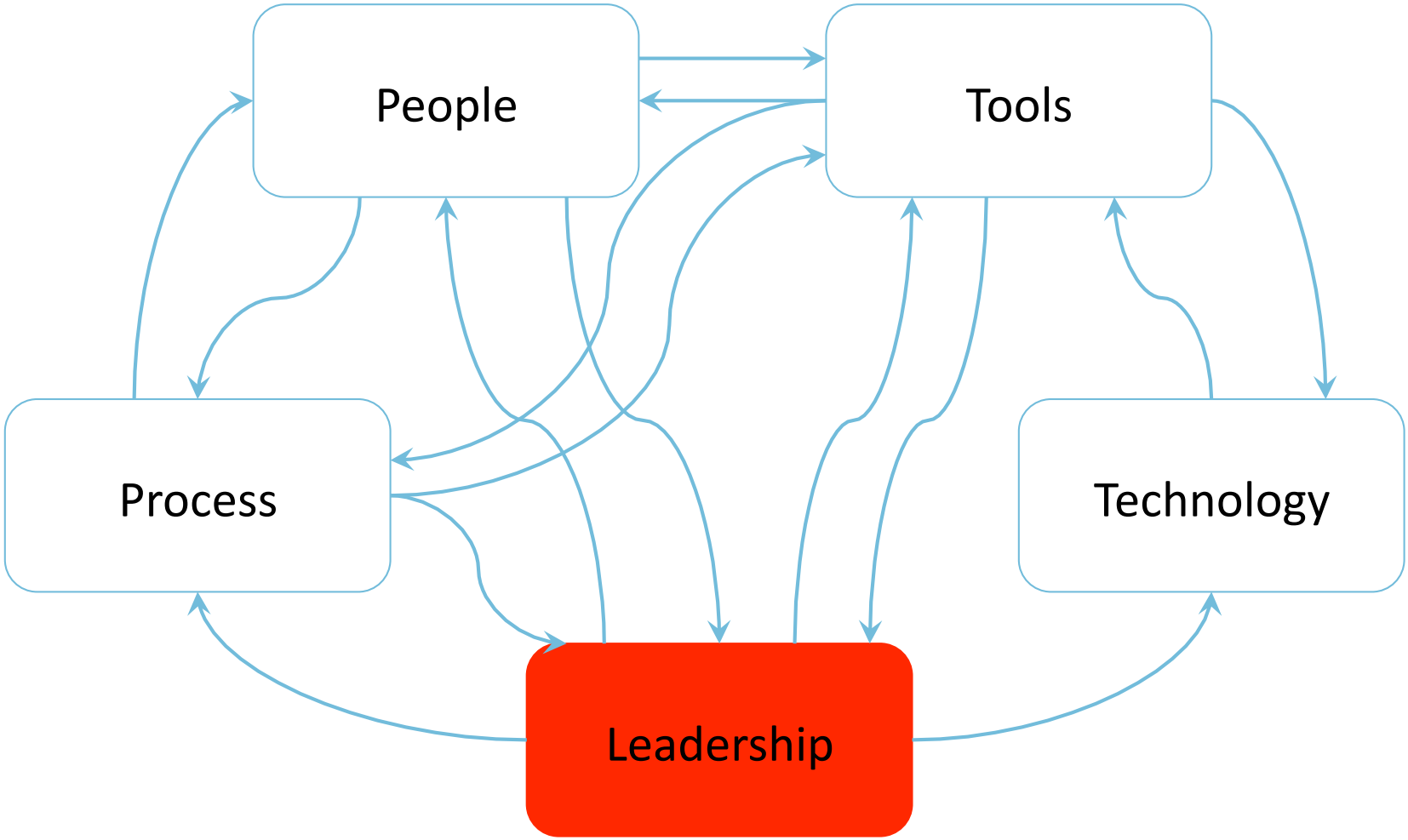


Our current system does not create effective Technical Leaders

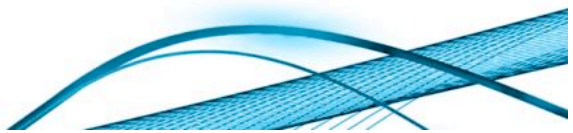
Problem Statement



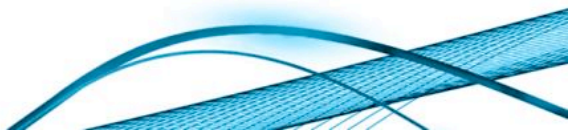
Software systems



Should we be worried?



Net Negative Producing Programmer

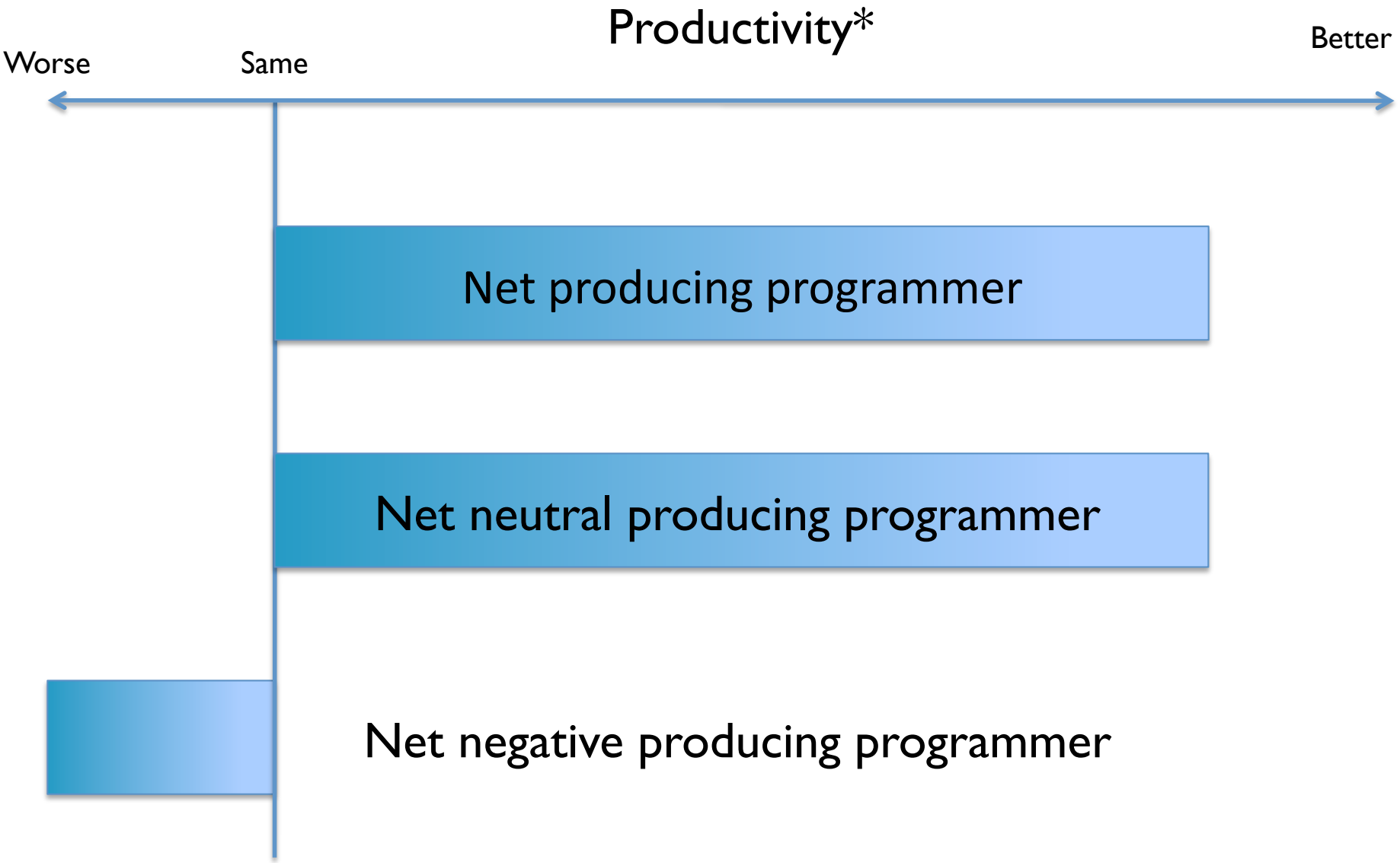


Define…

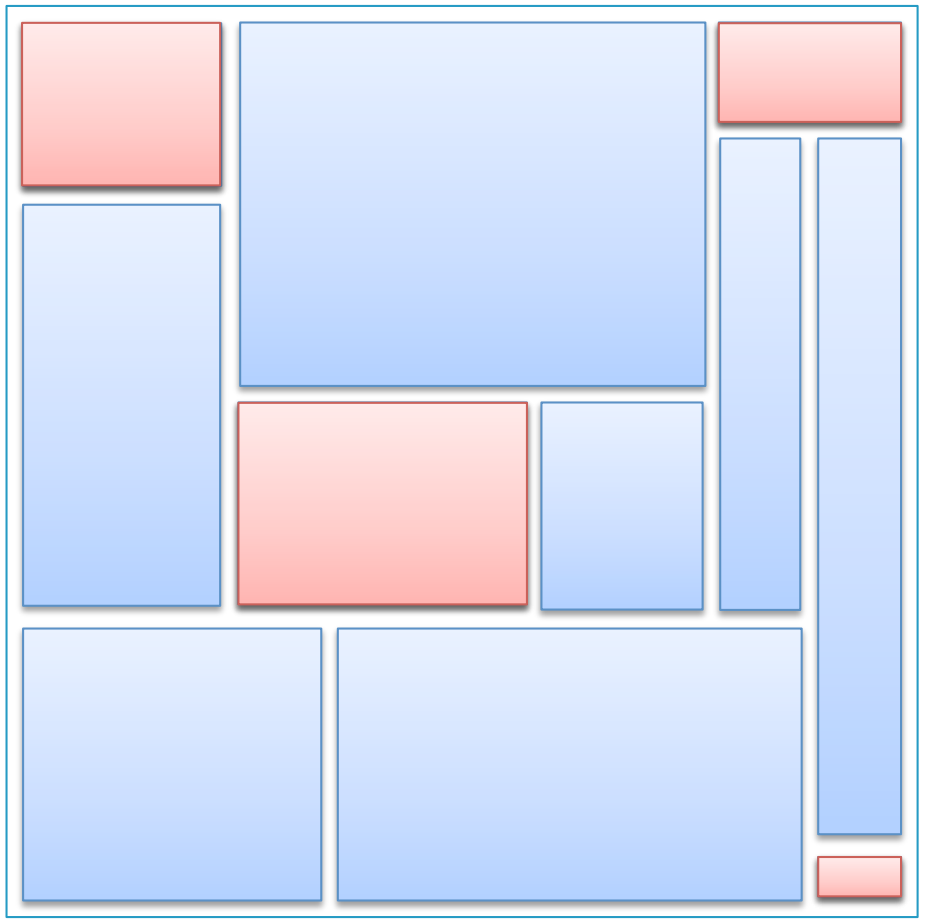
Net Negative Producing Programmer

(noun) [a person] who insert enough spoilage to exceed the value of their production

"The Net Negative Producing Programmer"
by G. Gordon Schulmeyer



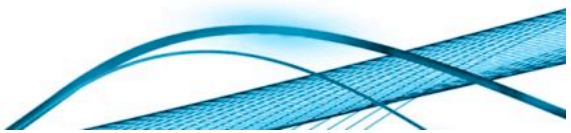
*If you can measure productivity

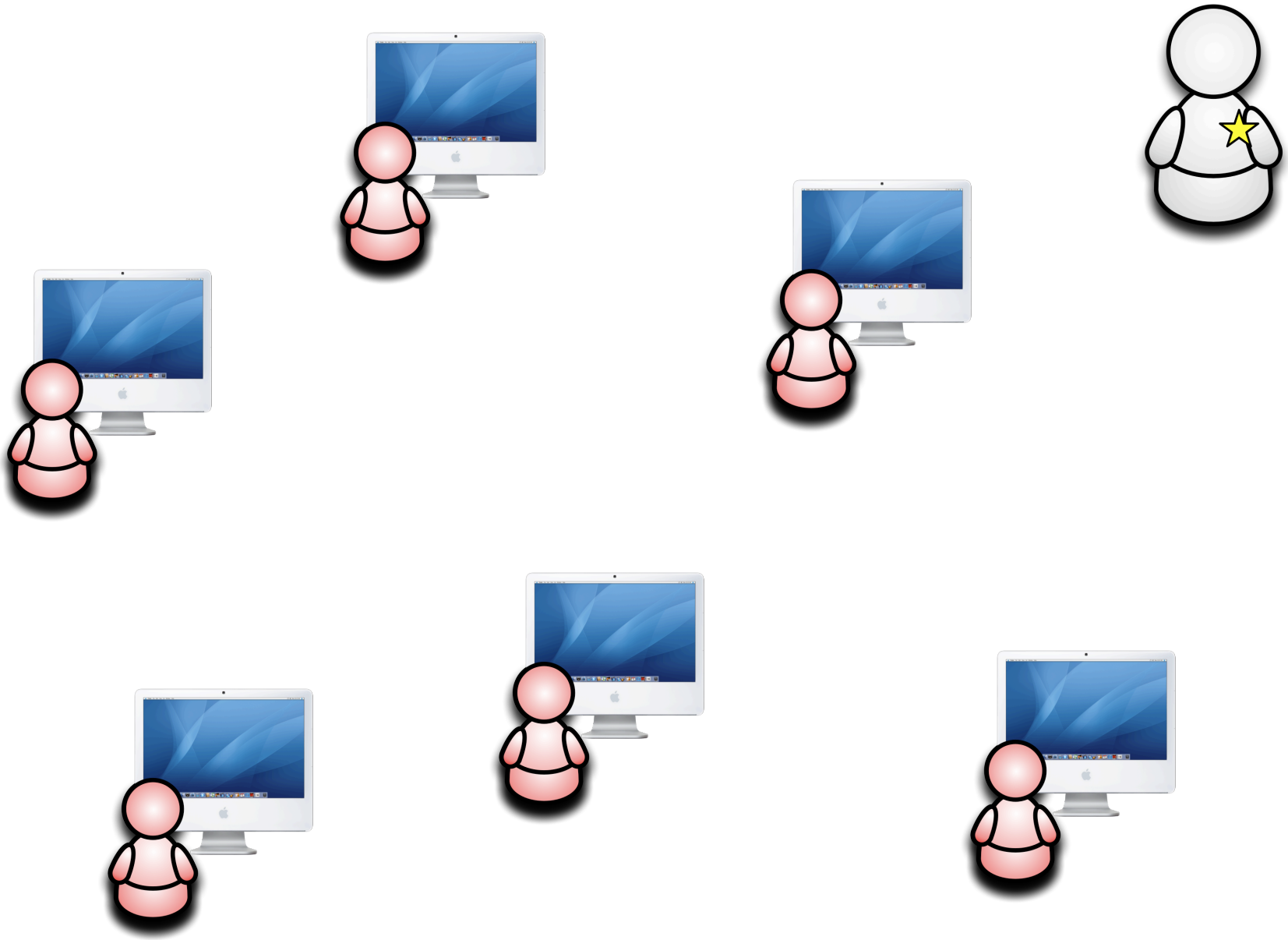


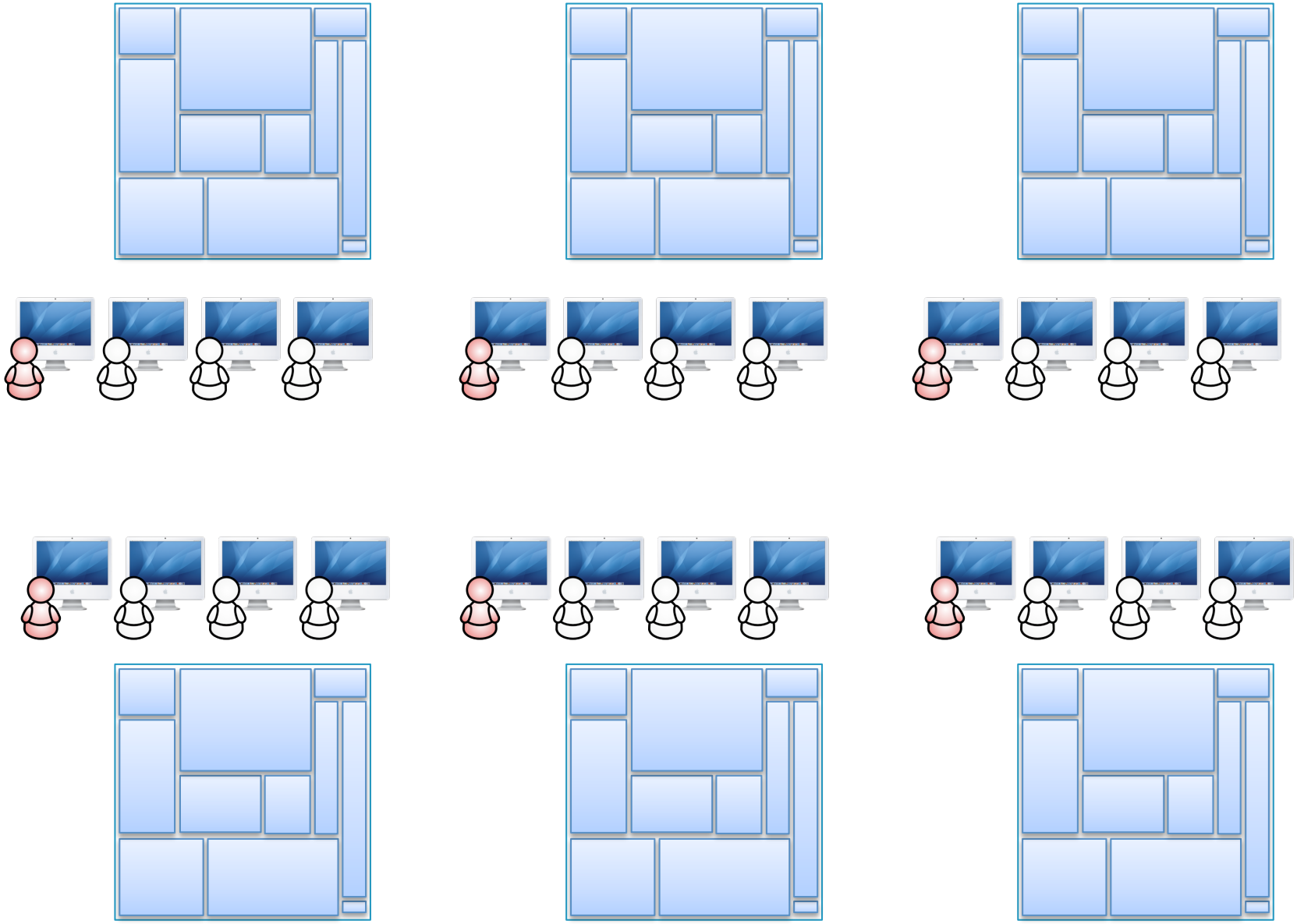
ThoughtWorks®

Net Negative Producing

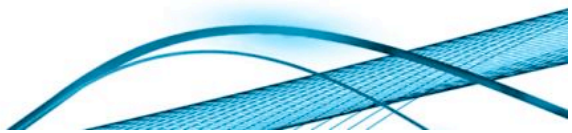
Technical Leader



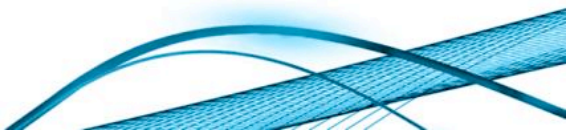


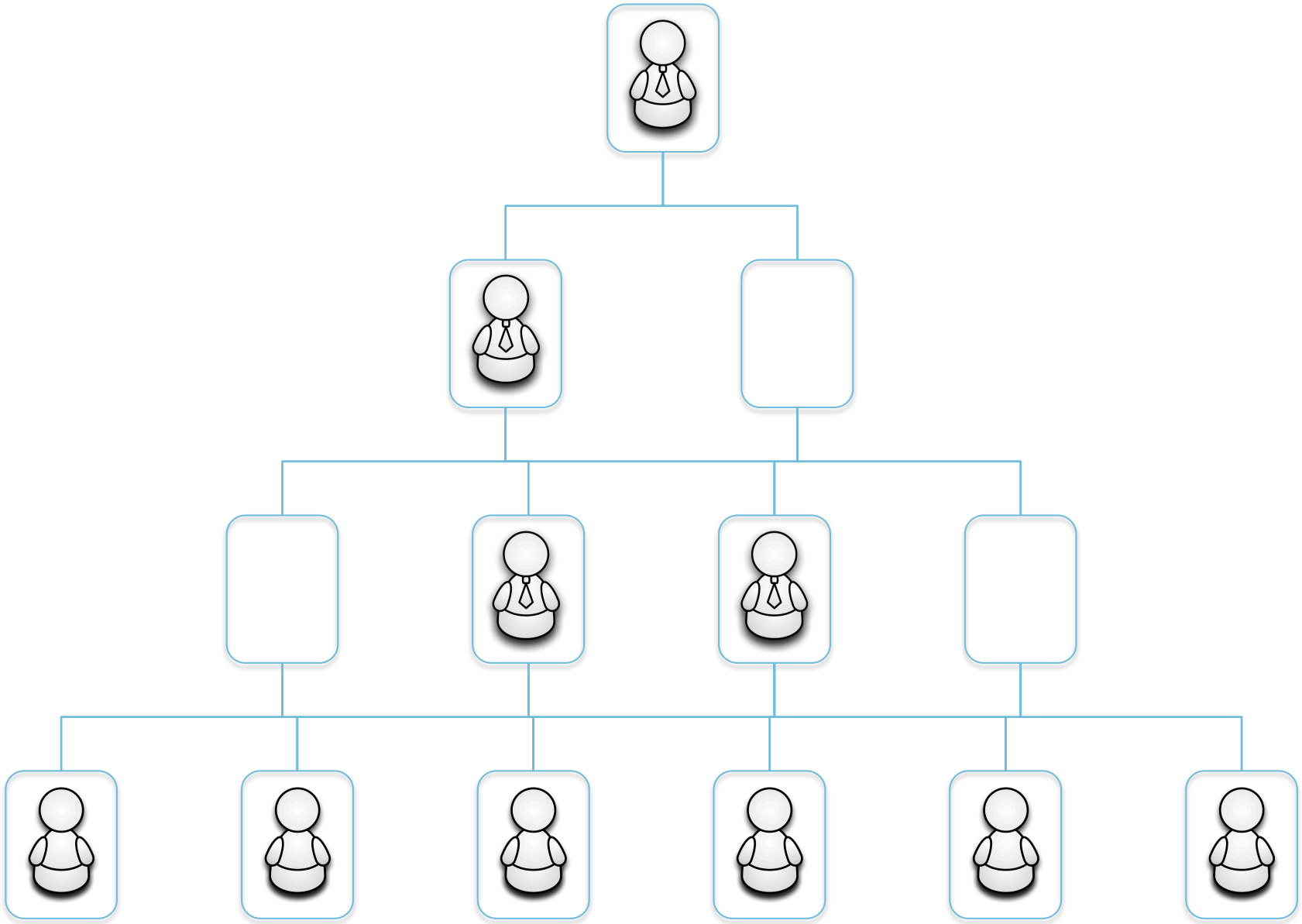


But how did they get there?



It’s all perfectly logical…

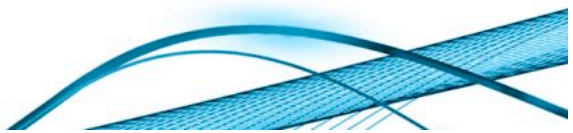




"In a hierarchy every employee tends to rise to his level of incompetence"

"The Peter Principle" (1969)

by Dr. Laurence J. Peter and Raymond Hull



Development

Implementing new functionality

Changing existing behaviour

Writing tests

Clean code

Challenging assumptions

Offering alternative solutions

Leadership

Ensuring everyone works towards a vision

Resolving conflict

Responsible for delivery

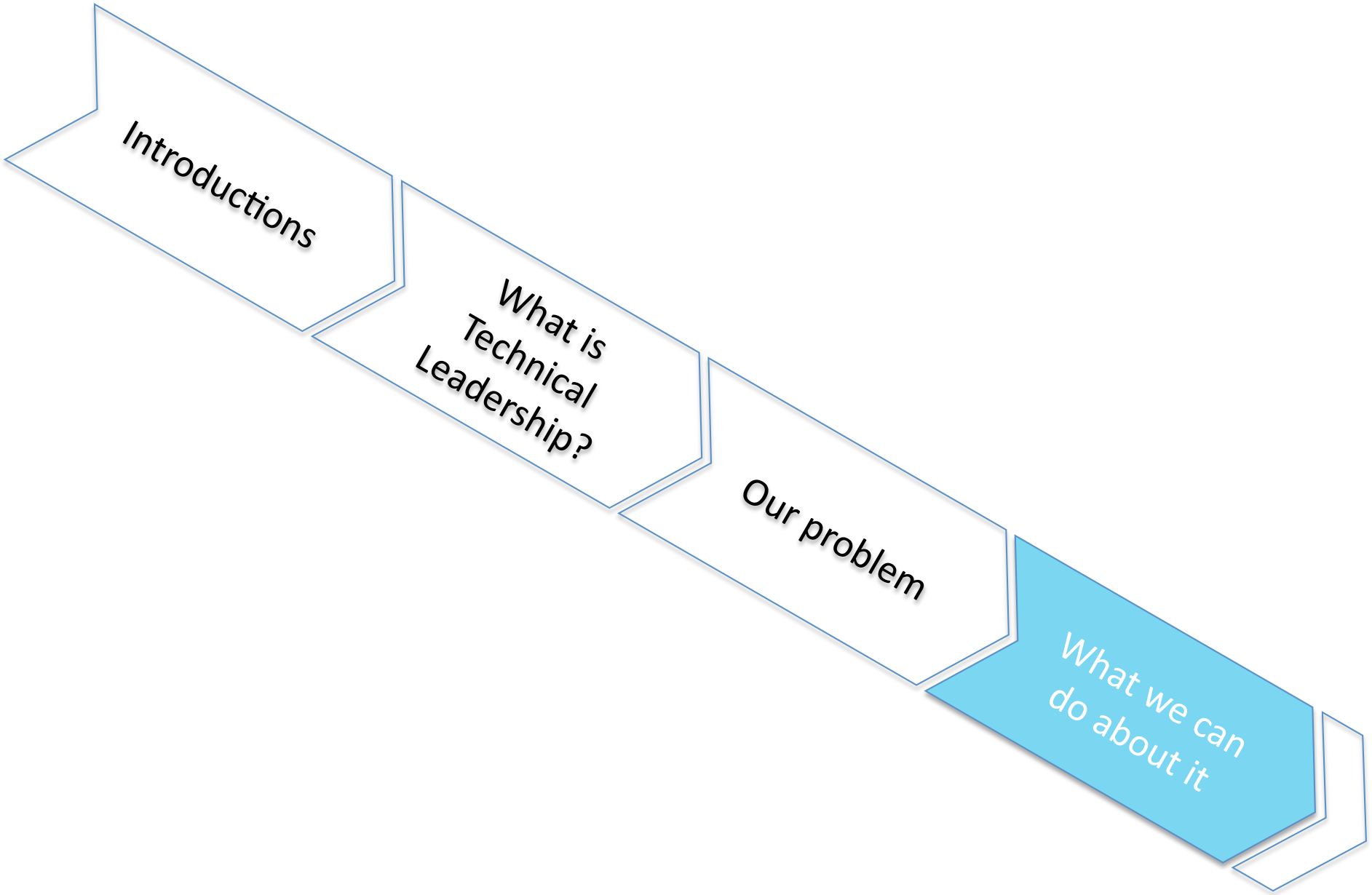
Growing talent

Story telling

Empowering everyone to make decisions

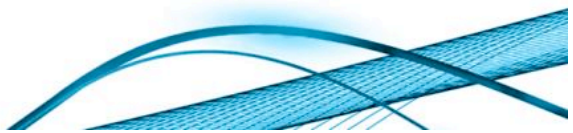
Creating a healthy team environment

ThoughtWorks®



ThoughtWorks®

Step 1: Recognise the different skillset



Development

Implementing new functionality

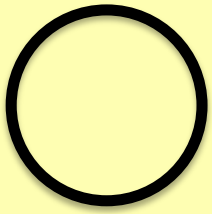
Changing existing behaviour

Writing tests

Clean code

Challenging assumptions

Offering alternative solutions



Leadership

Ensuring everyone works towards a vision

Resolving conflict

Responsible for delivery

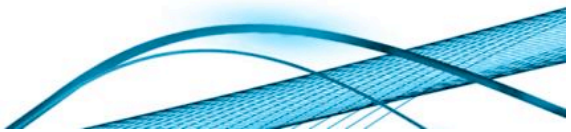
Growing talent

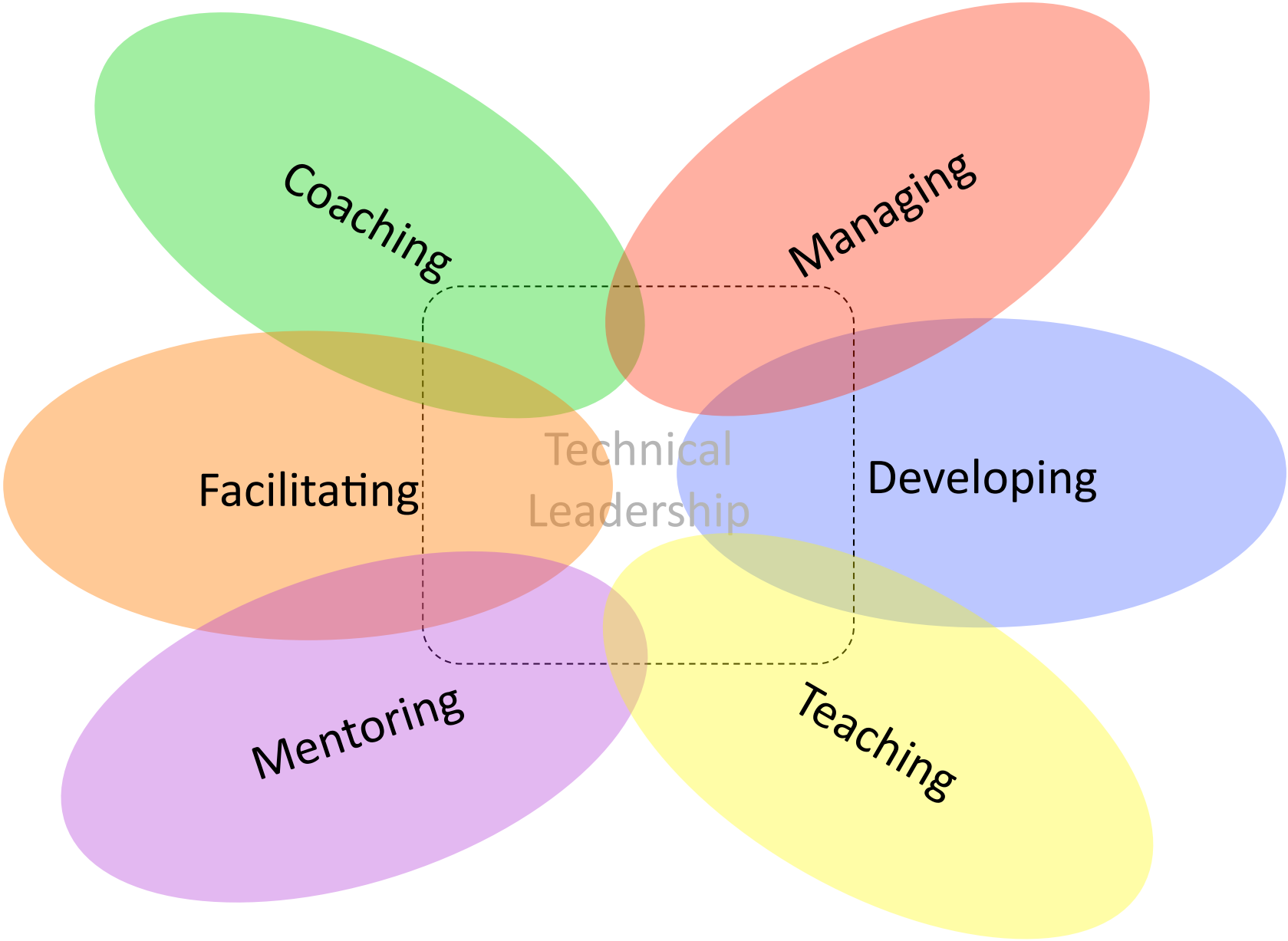
Story telling

Empowering everyone to make decisions

Creating a healthy team environment

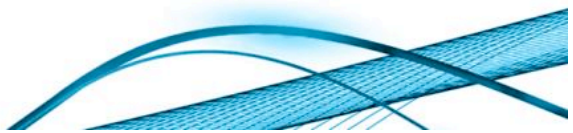
ThoughtWorks®



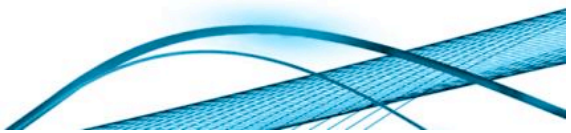

Coaching
Managing
Facilitating
Technical
Leadership
Developing
Mentoring
Teaching

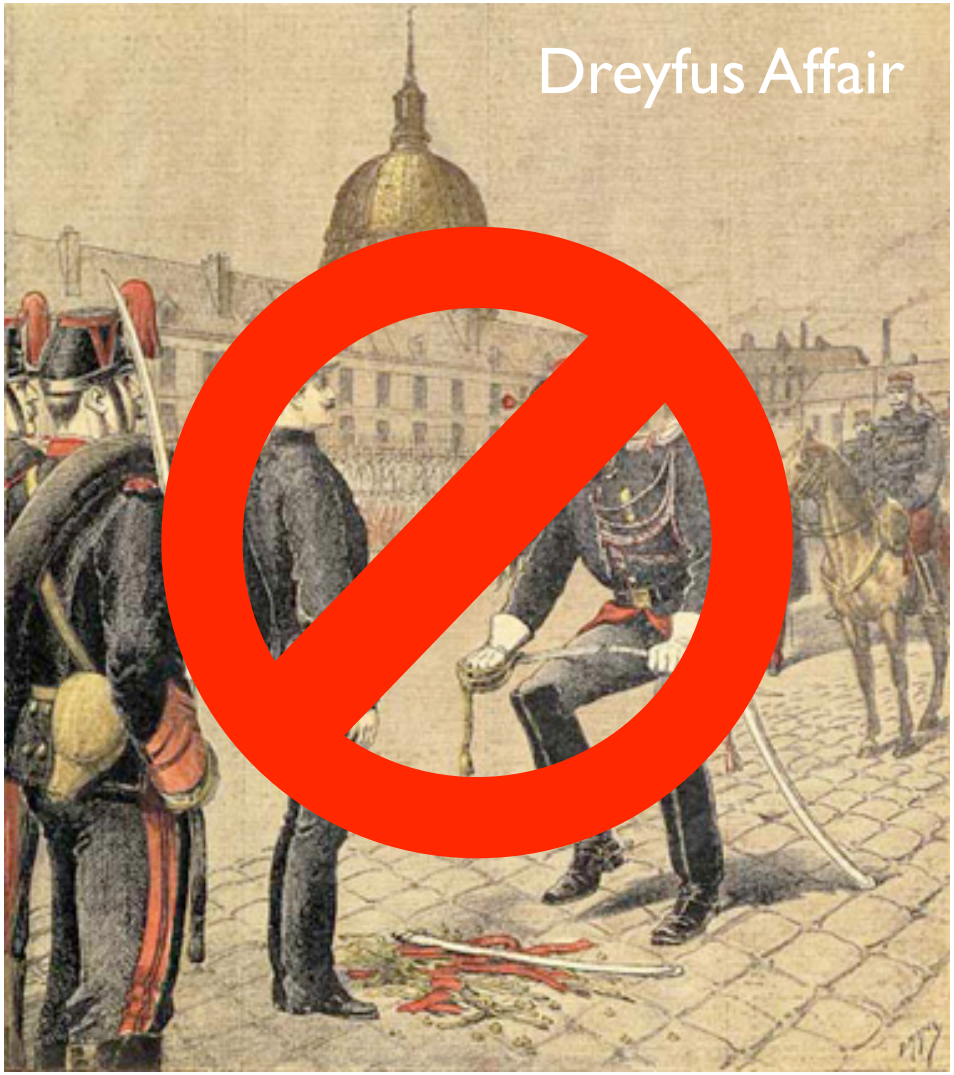
Step 1: Recognise the different skillset

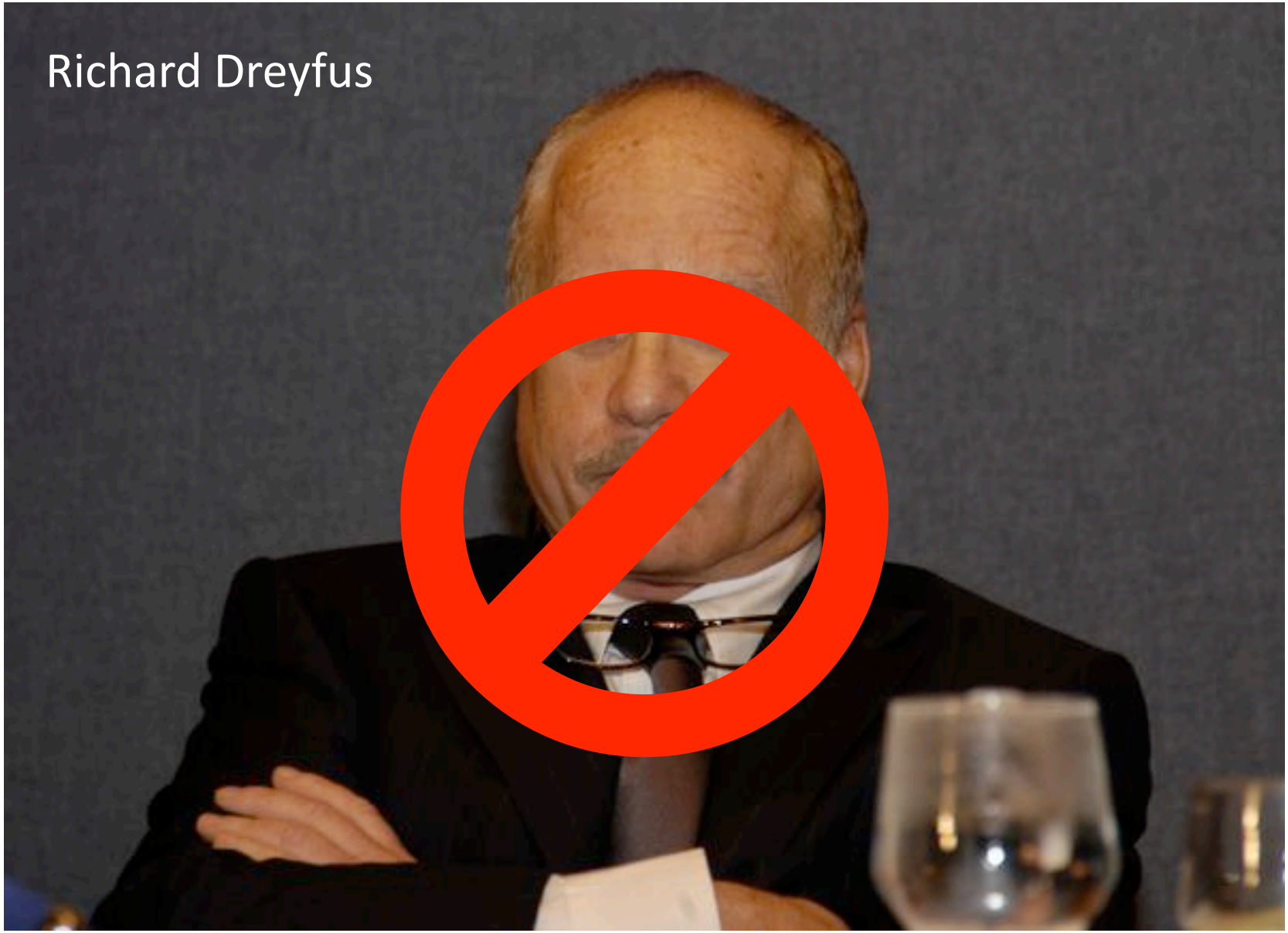
Step 2: Develop competence

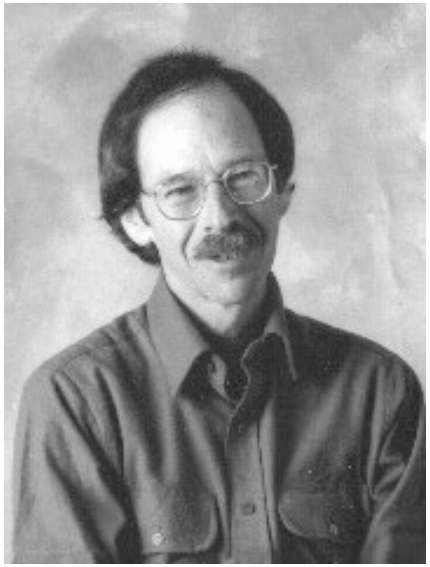


Dreyfus model of skills acquisition



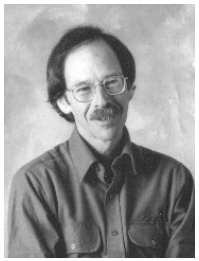




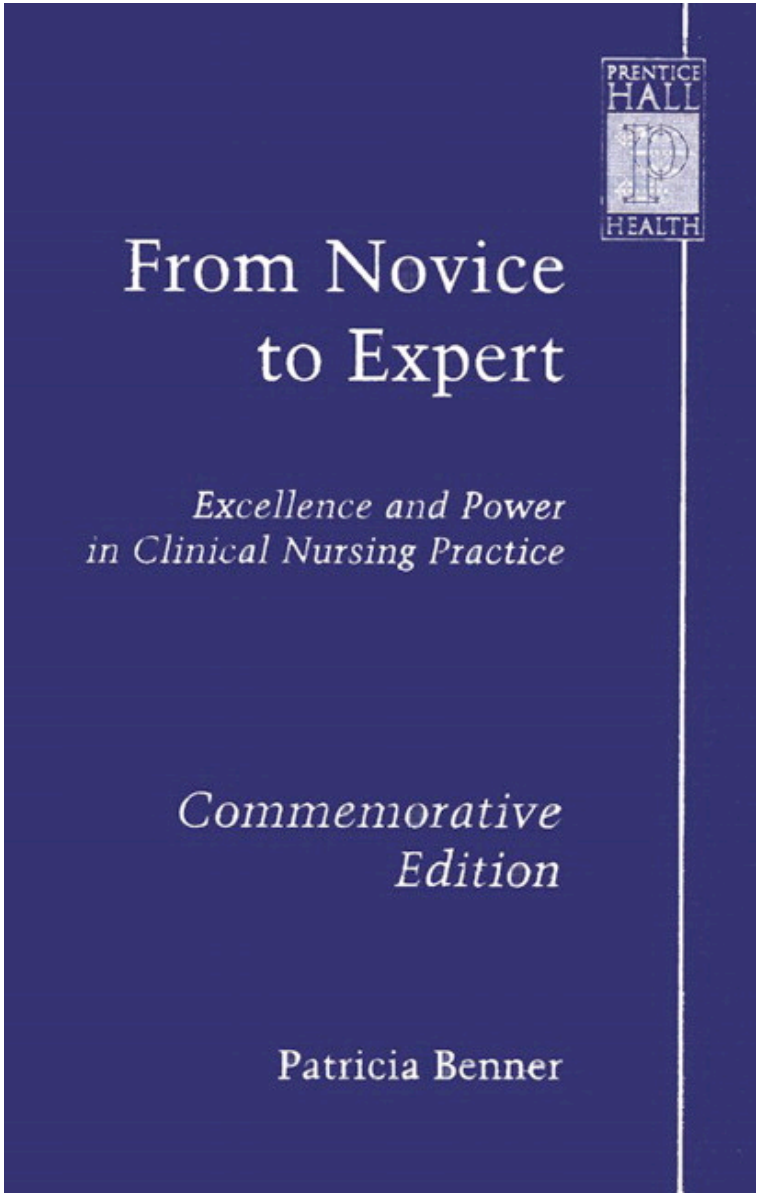


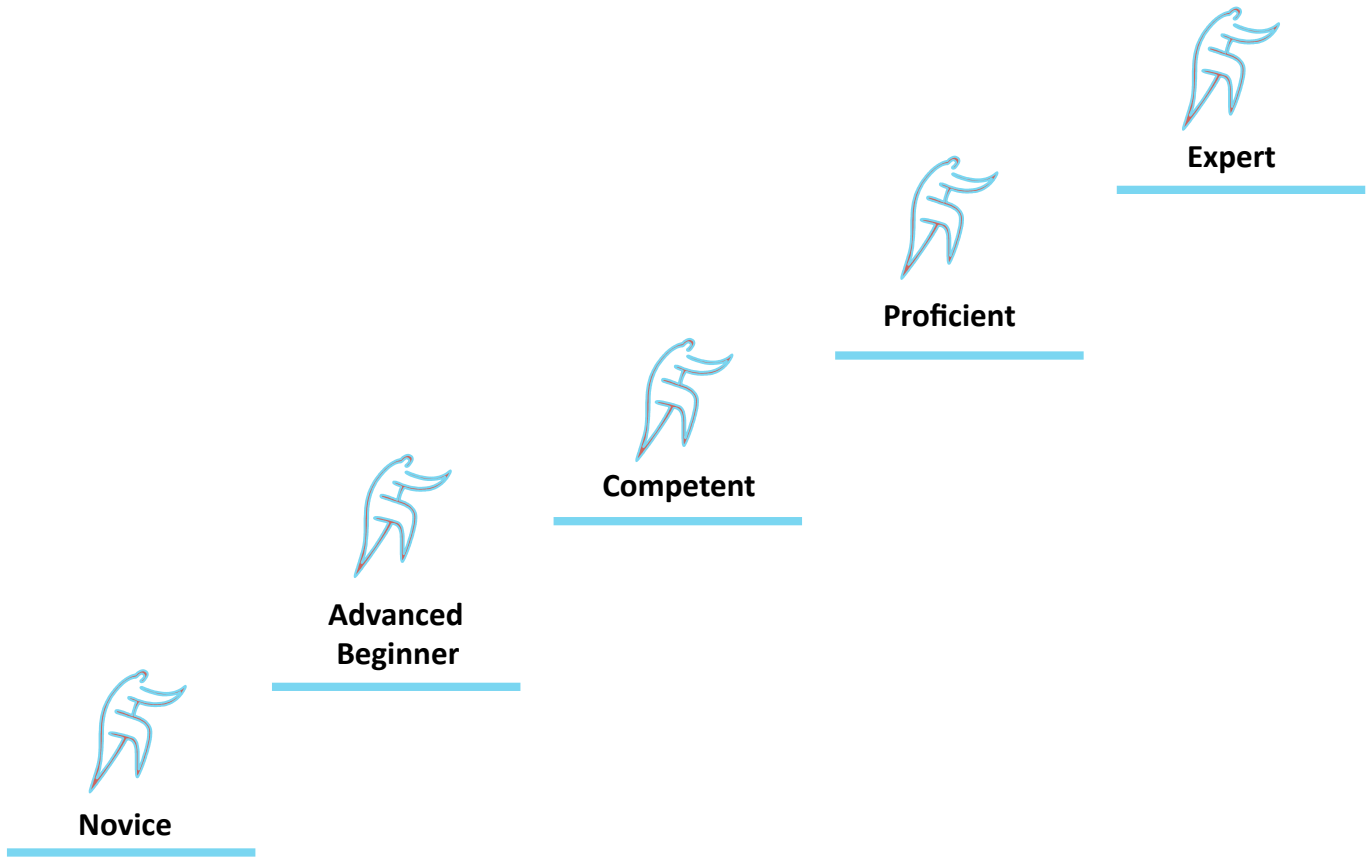
Stuart Dreyfus

Stuart Dreyfus



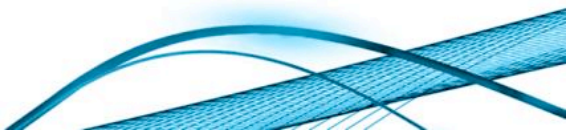
Paper: A Five-Stage Model of the Mental Activities Involved in Directed Skill Acquisition





Don't put a

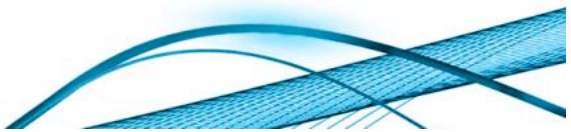
Novice in control



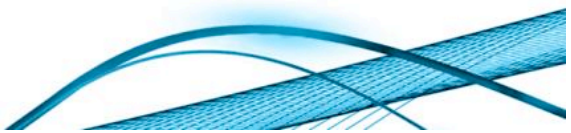
Remember this?

Development

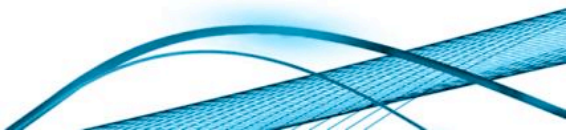
Leadership



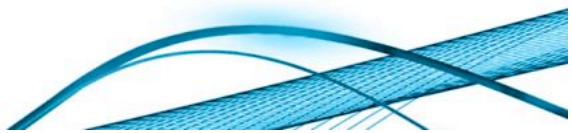
Pair ing ~~Programming~~



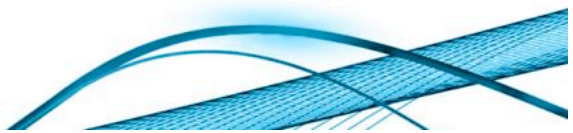
l:ls



Shadowing



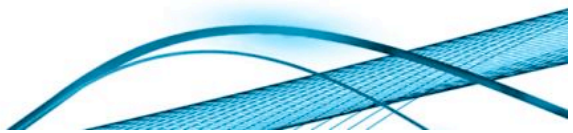
Safe environments



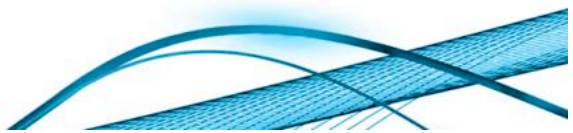
Step 1: Recognise the different skillset

Step 2: Develop competence

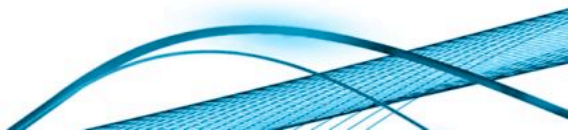
Step 4: Profit! (really)



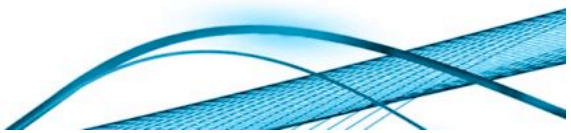
Learning as a team



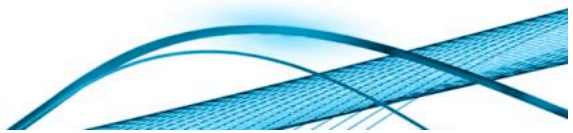
Celebrate success



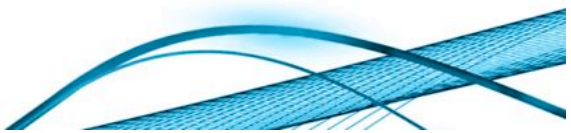
Opportunities to contribute

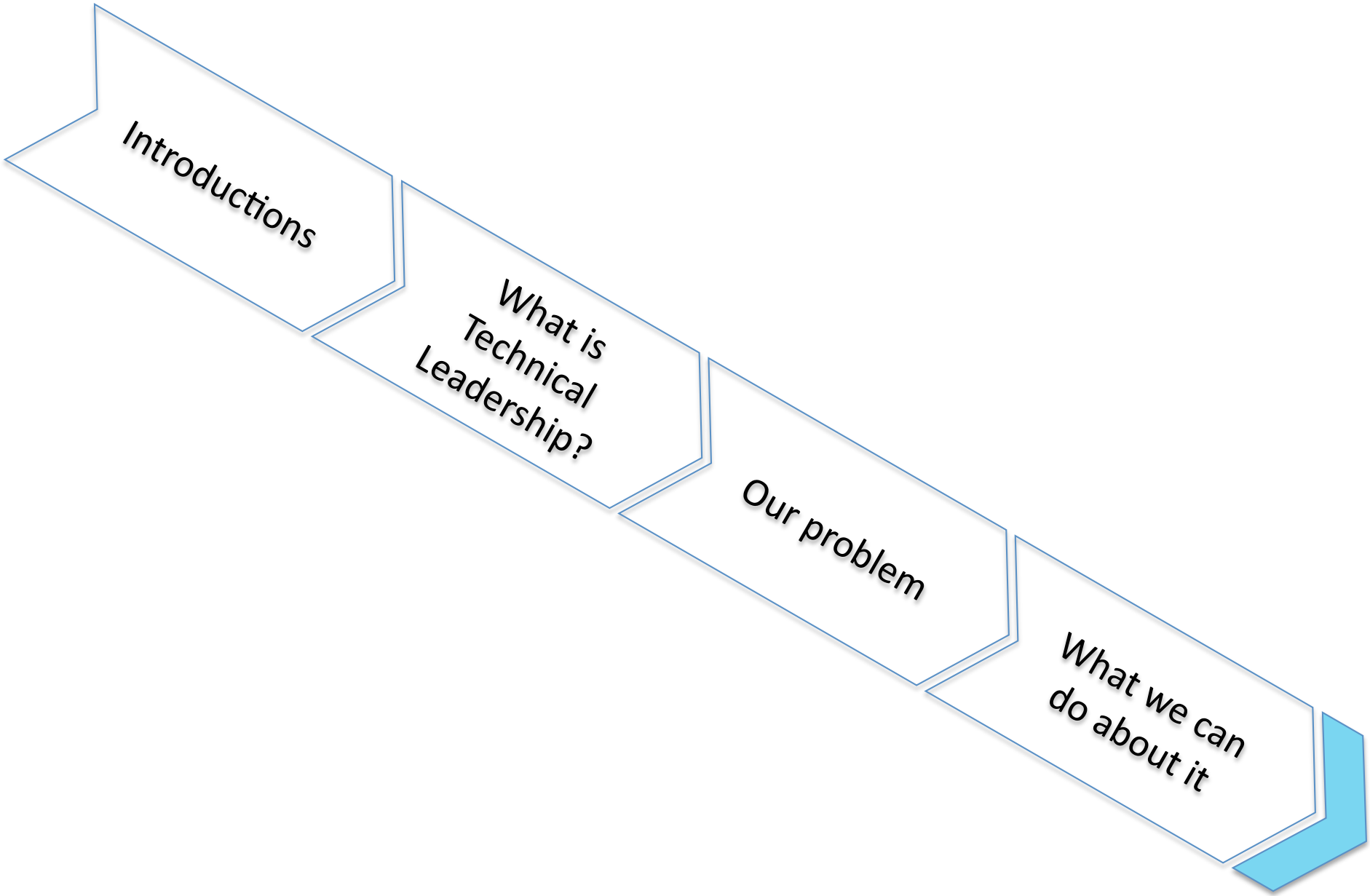


Everyone is safe to fail
(lead by humility)

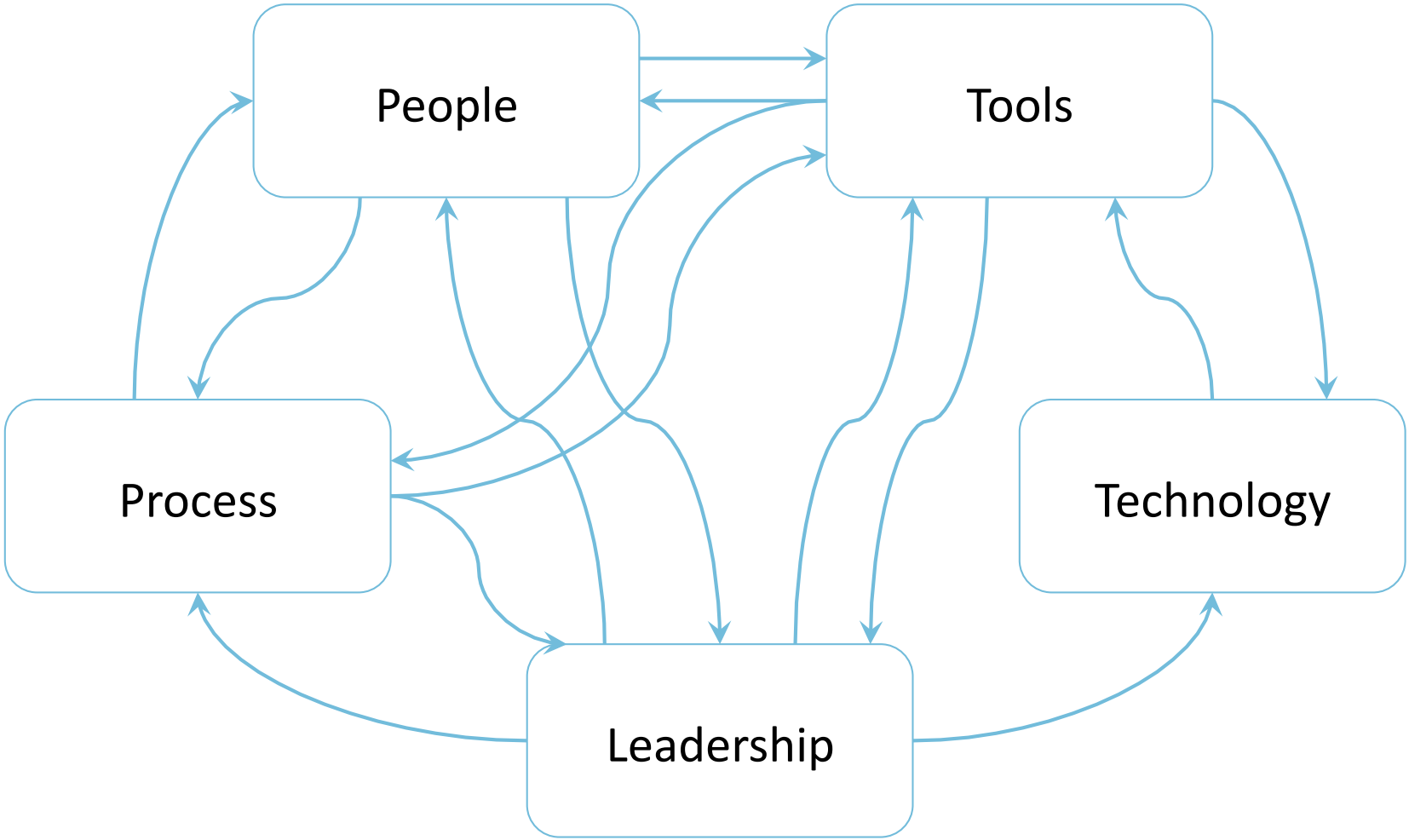


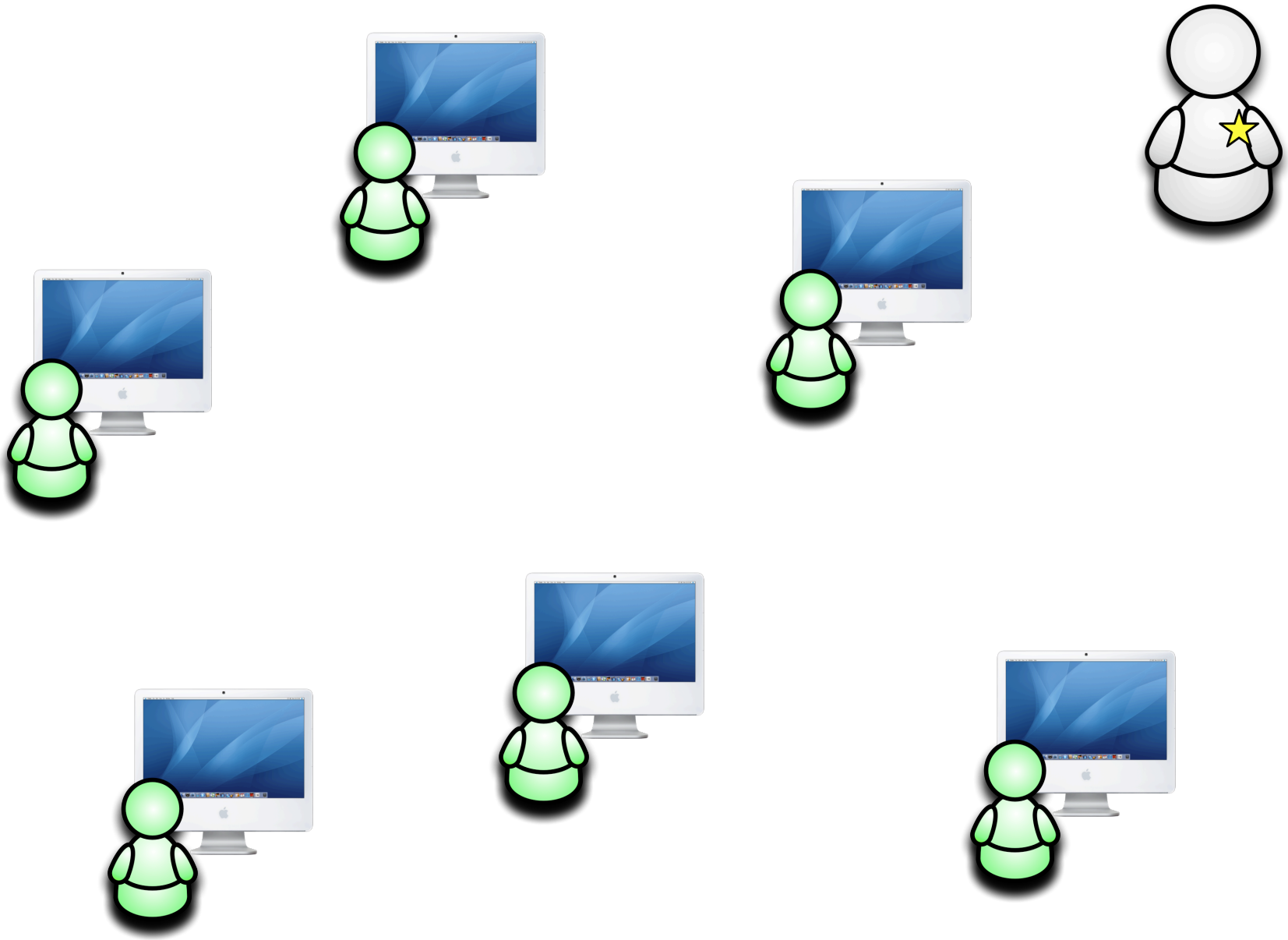
Conflicts resolved openly



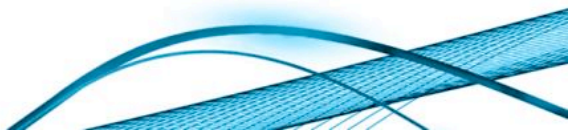

Introductions
What is Technical Leadership?
Our problem
What we can do about it

Software systems



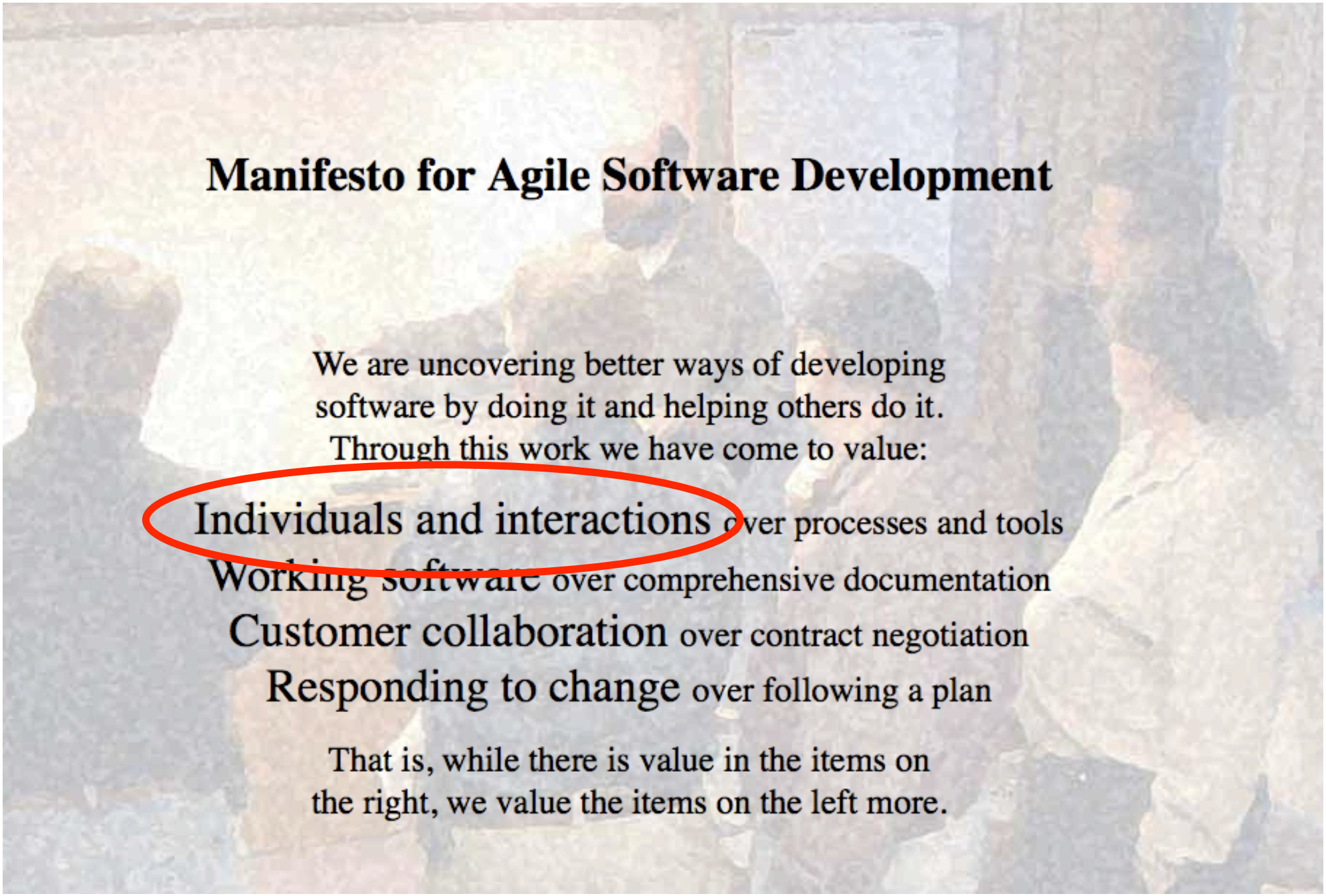


Closing notes...



"...it is critical to acknowledge the impact of individuals and teams on end results"

Source: QCon London recursive reference


Manifesto for Agile Software Development
We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:
Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan
That is, while there is value in the items on
the right, we value the items on the left more.

Questions?

patrick.kua@thoughtworks.com

Twitter: @patkua