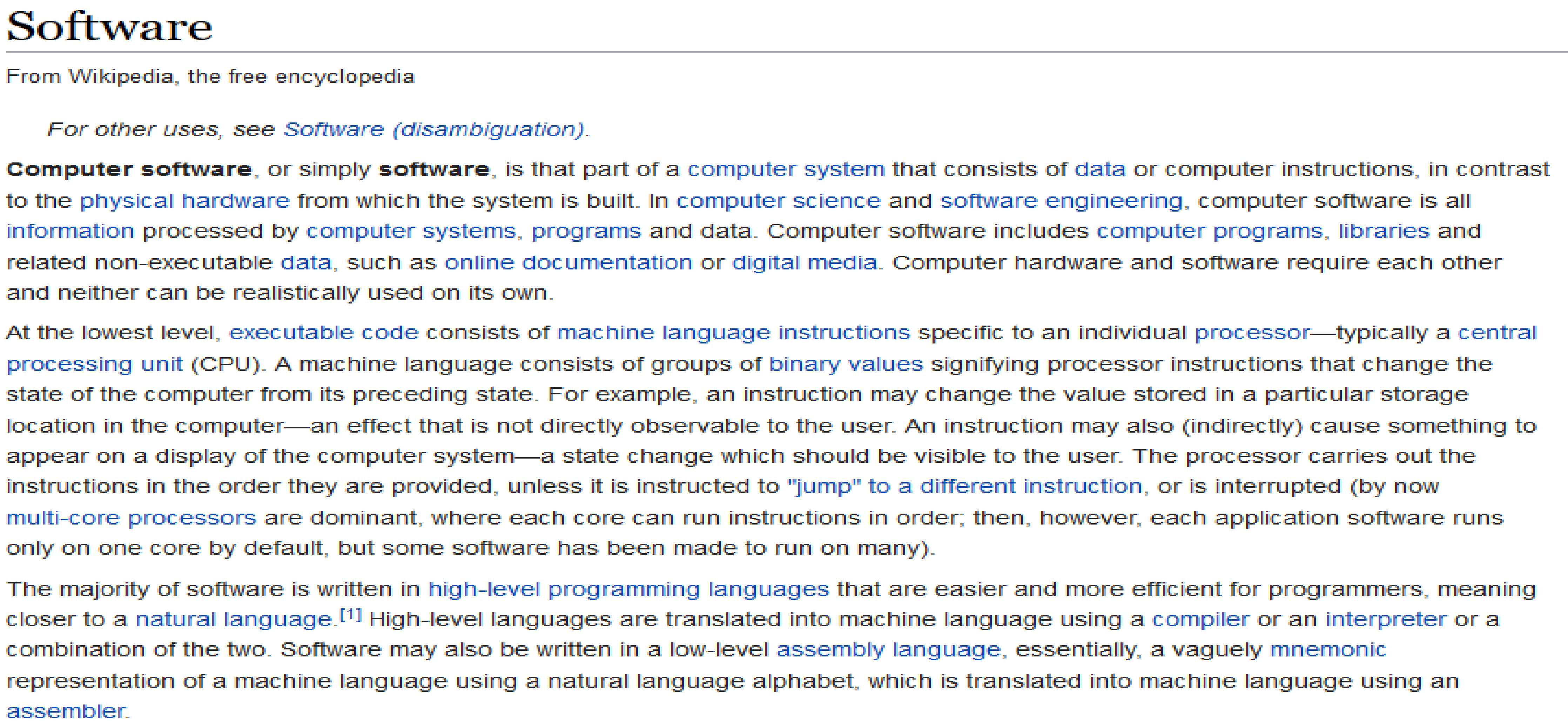
Hippocampus.io
consulting ventures

Doing S/W Right OR doing the Right S/W?

Christos Lytras

Managing Partner

What Is S/W



Software

From Wikipedia, the free encyclopedia

For other uses, see Software (disambiguation).

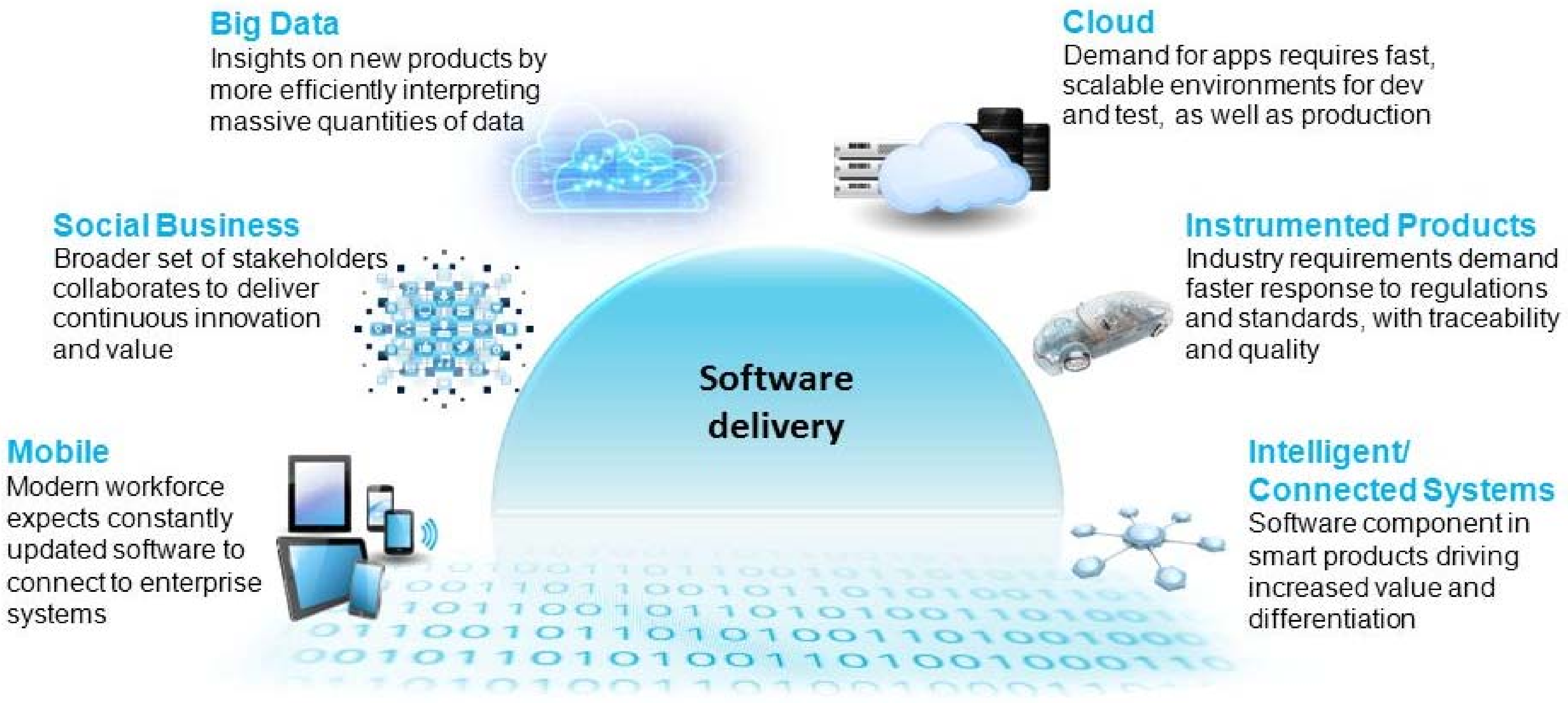
Computer software, or simply **software**, is that part of a computer system that consists of data or computer instructions, in contrast to the physical hardware from which the system is built. In computer science and software engineering, computer software is all information processed by computer systems, programs and data. Computer software includes computer programs, libraries and related non-executable data, such as online documentation or digital media. Computer hardware and software require each other and neither can be realistically used on its own.

At the lowest level, executable code consists of machine language instructions specific to an individual processor—typically a central processing unit (CPU). A machine language consists of groups of binary values signifying processor instructions that change the state of the computer from its preceding state. For example, an instruction may change the value stored in a particular storage location in the computer—an effect that is not directly observable to the user. An instruction may also (indirectly) cause something to appear on a display of the computer system—a state change which should be visible to the user. The processor carries out the instructions in the order they are provided, unless it is instructed to "jump" to a different instruction, or is interrupted (by now multi-core processors are dominant, where each core can run instructions in order; then, however, each application software runs only on one core by default, but some software has been made to run on many).

The majority of software is written in high-level programming languages that are easier and more efficient for programmers, meaning closer to a natural language.[1] High-level languages are translated into machine language using a compiler or an interpreter or a combination of the two. Software may also be written in a low-level assembly language, essentially, a vaguely mnemonic representation of a machine language using a natural language alphabet, which is translated into machine language using an assembler.

Constant Change Is The New Normal

S/W @ the heart of 21st centure Trends



Big Data
Insights on new products by more efficiently interpreting massive quantities of data

Cloud
Demand for apps requires fast, scalable environments for dev and test, as well as production

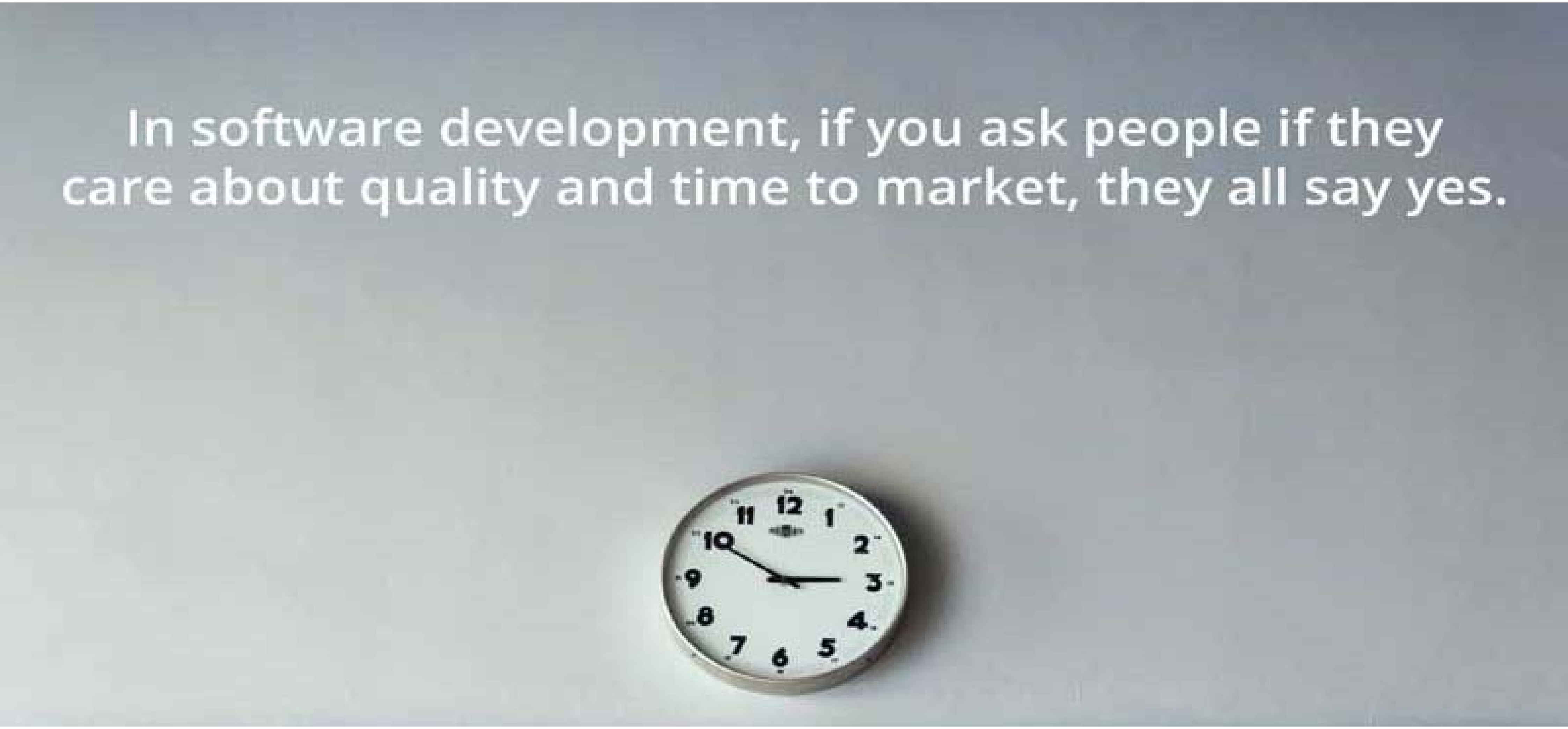
Social Business
Broader set of stakeholders collaborates to deliver continuous innovation and value

Instrumented Products
Industry requirements demand faster response to regulations and standards, with traceability and quality

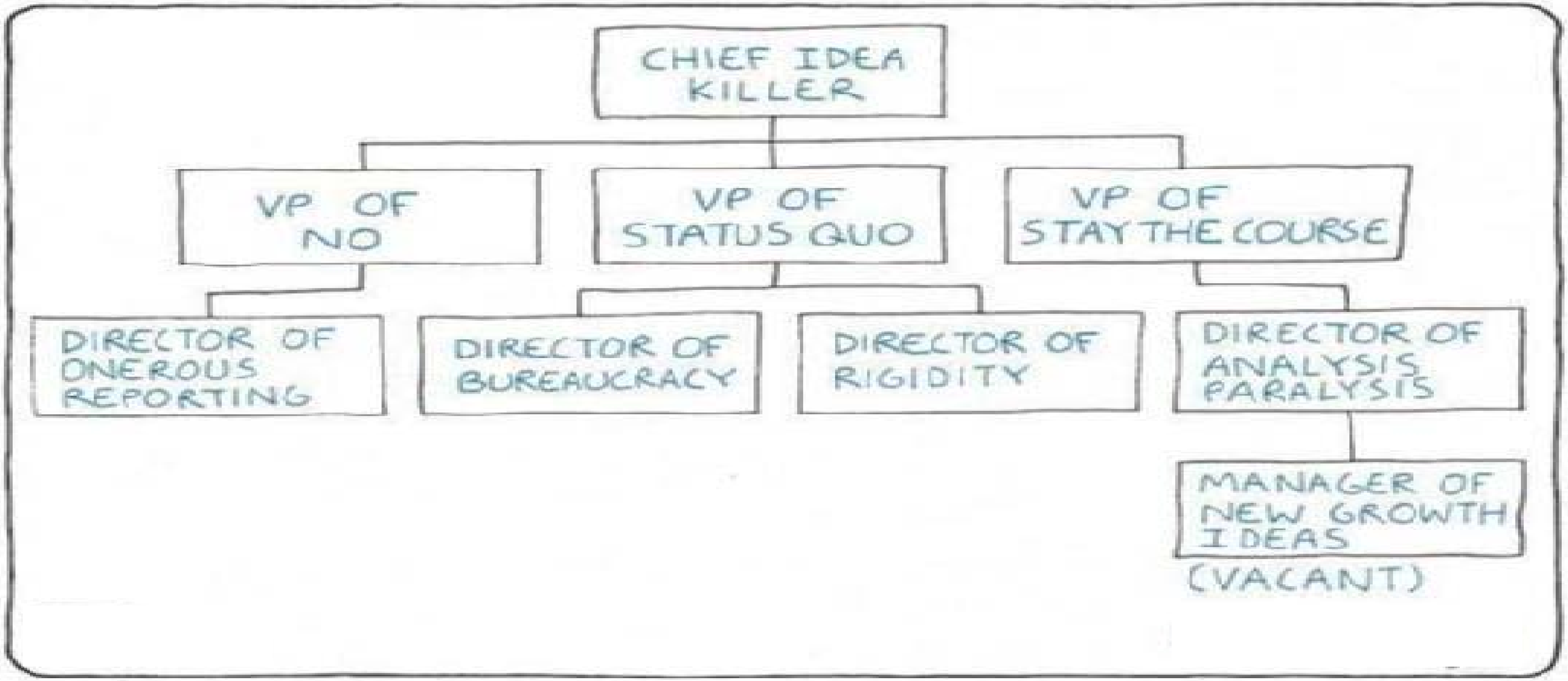
Mobile
Modern workforce expects constantly updated software to connect to enterprise systems

Intelligent/ Connected Systems
Software component in smart products driving increased value and differentiation

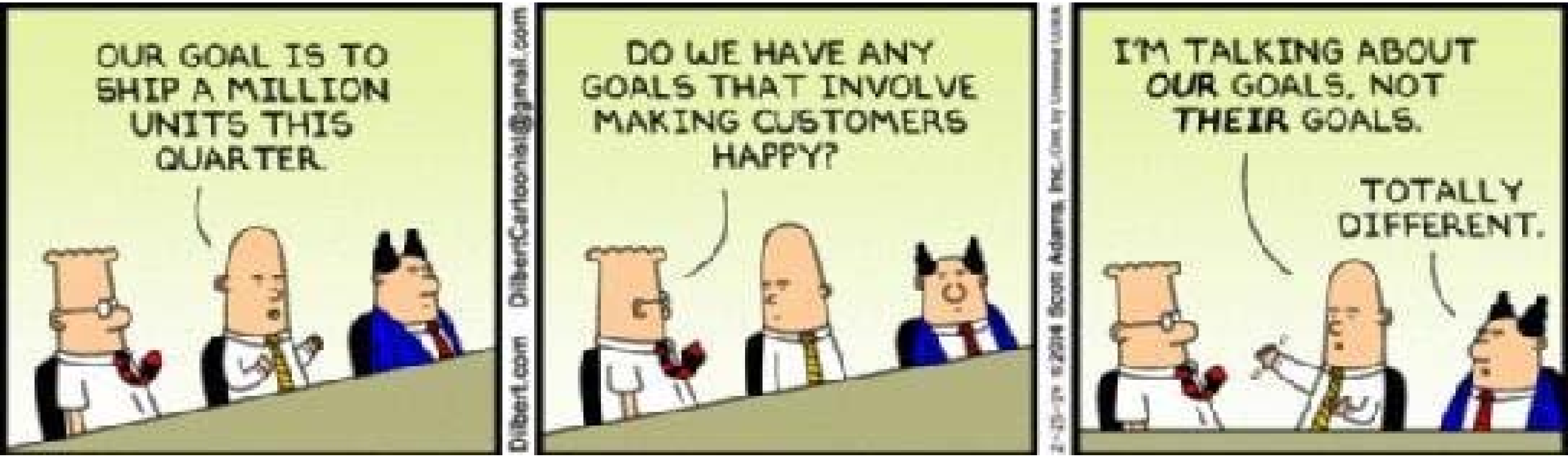
What does the Customer Want?


In software development, if you ask people if they care about quality and time to market, they all say yes.

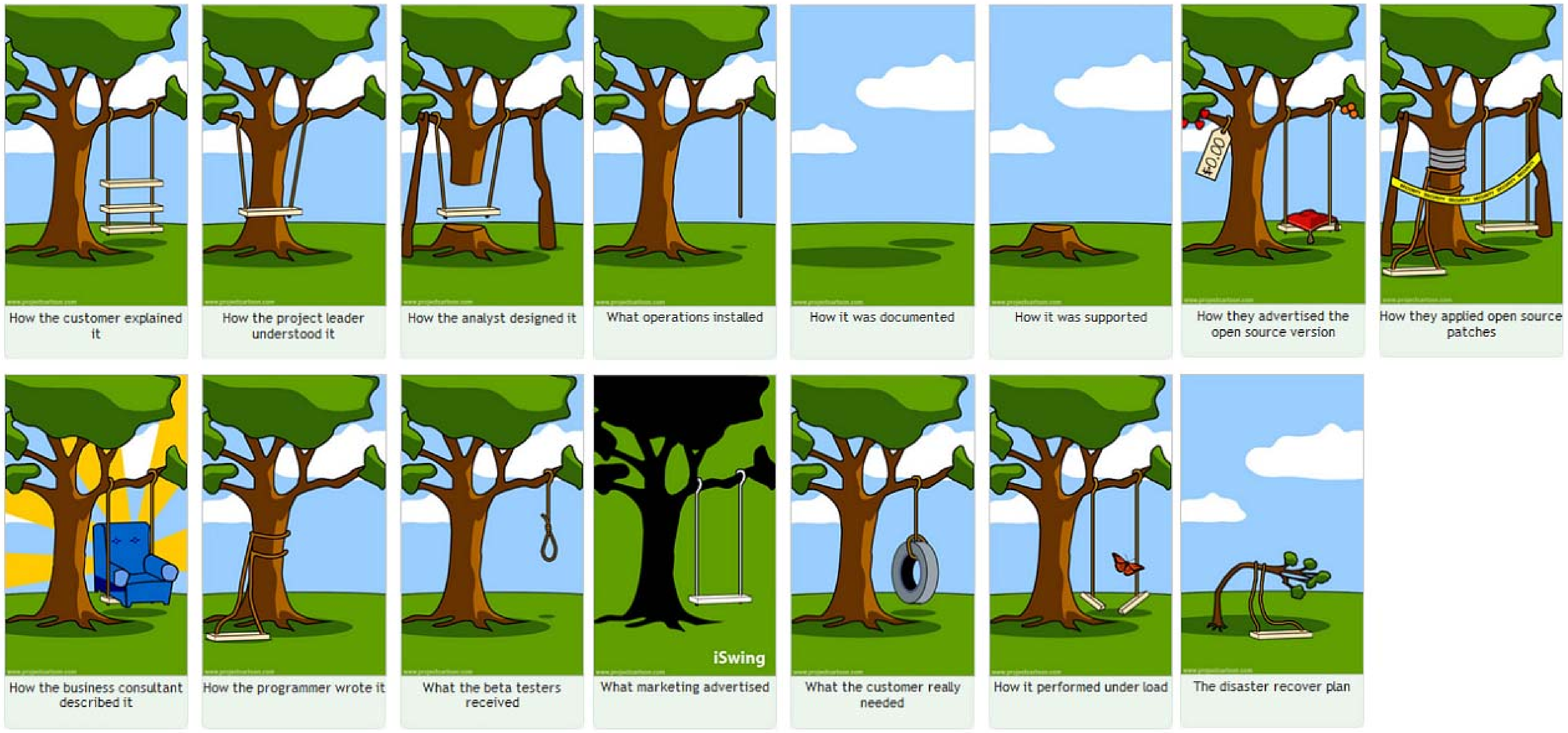
Addressing the Customer Pain



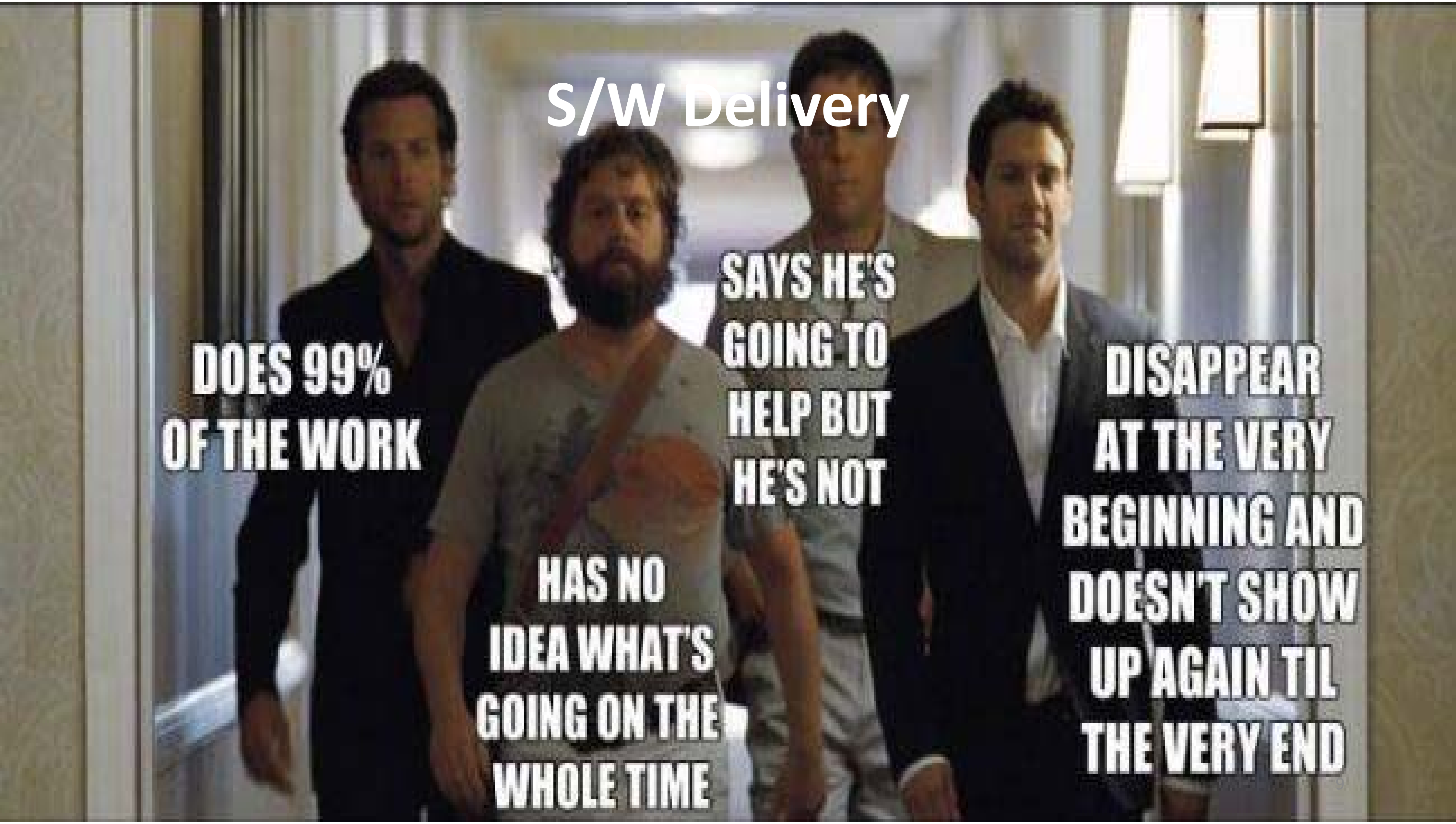
Solving the Customers Problem


OUR GOAL IS TO SHIP A MILLION UNITS THIS QUARTER.
DO WE HAVE ANY GOALS THAT INVOLVE MAKING CUSTOMERS HAPPY?
I'M TALKING ABOUT OUR GOALS, NOT THEIR GOALS.
TOTALLY DIFFERENT.
Dilbert.com DilbertCartoonist@gmail.com
© 2014 Scott Adams, Inc.

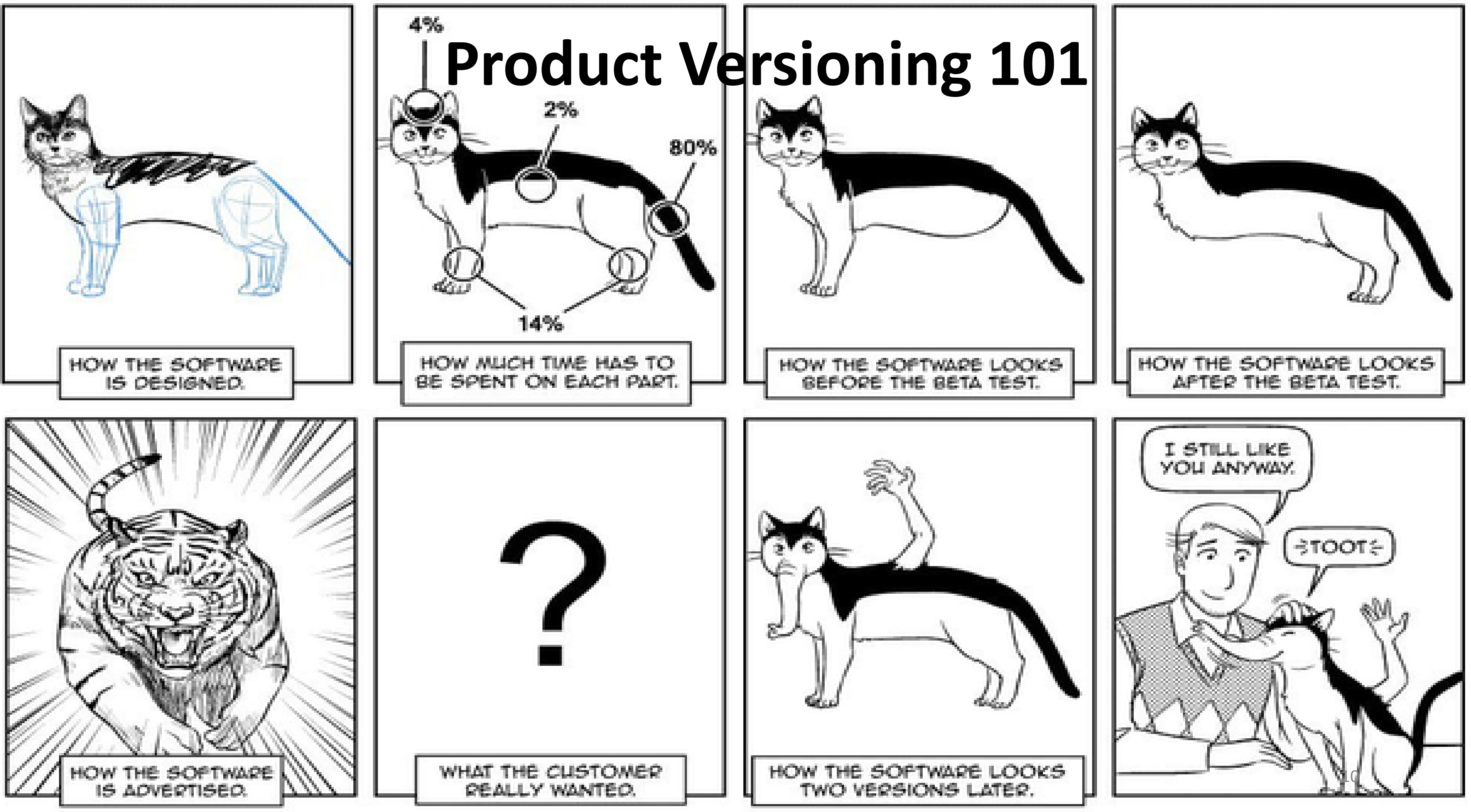
Customer needs Definition


How the customer explained it
How the project leader understood it
How the analyst designed it
What operations installed
How it was documented
How it was supported
$0.00
How they advertised the open source version
How they applied open source patches
How the business consultant described it
How the programmer wrote it
What the beta testers received
iSwing
What marketing advertised
What the customer really needed
How it performed under load
The disaster recover plan

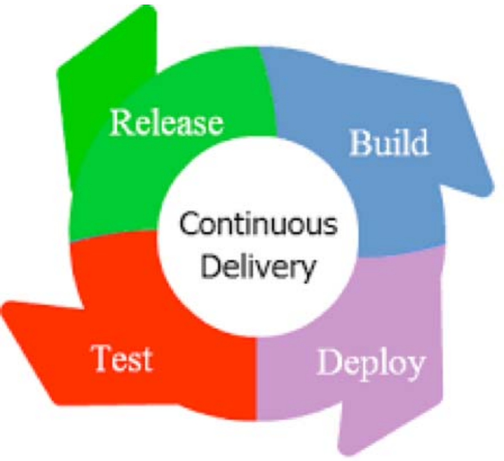
S/W Delivery

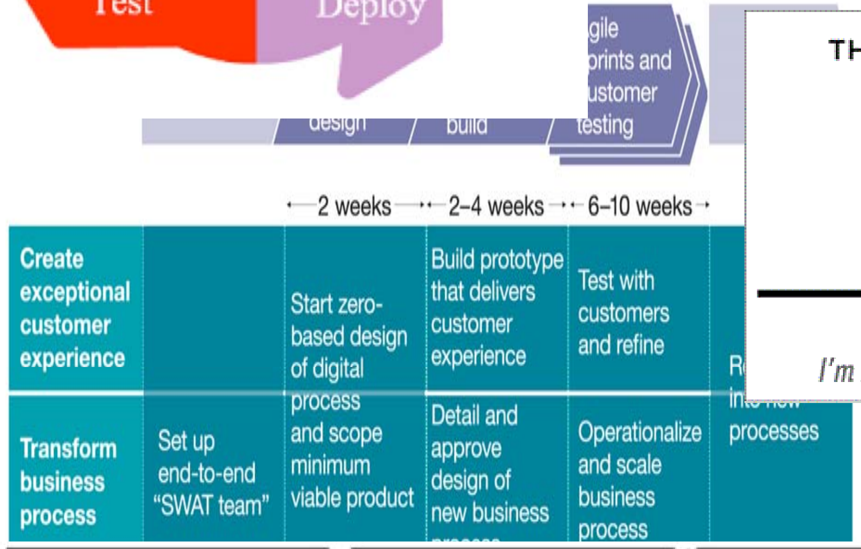

DOES 99% OF THE WORK
HAS NO IDEA WHAT'S GOING ON THE WHOLE TIME
SAYS HE'S GOING TO HELP BUT HE'S NOT
DISAPPEAR AT THE VERY BEGINNING AND DOESN'T SHOW UP AGAIN TIL THE VERY END

Product Versioning 101

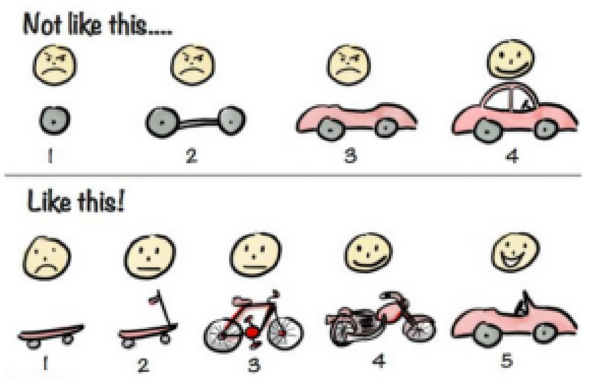

HOW THE SOFTWARE IS DESIGNED.
4%
2%
80%
14%
HOW MUCH TIME HAS TO BE SPENT ON EACH PART.
HOW THE SOFTWARE LOOKS BEFORE THE BETA TEST.
HOW THE SOFTWARE LOOKS AFTER THE BETA TEST.
HOW THE SOFTWARE IS ADVERTISED.
?
WHAT THE CUSTOMER REALLY WANTED.
HOW THE SOFTWARE LOOKS TWO VERSIONS LATER.
I STILL LIKE YOU ANYWAY.
TOOT

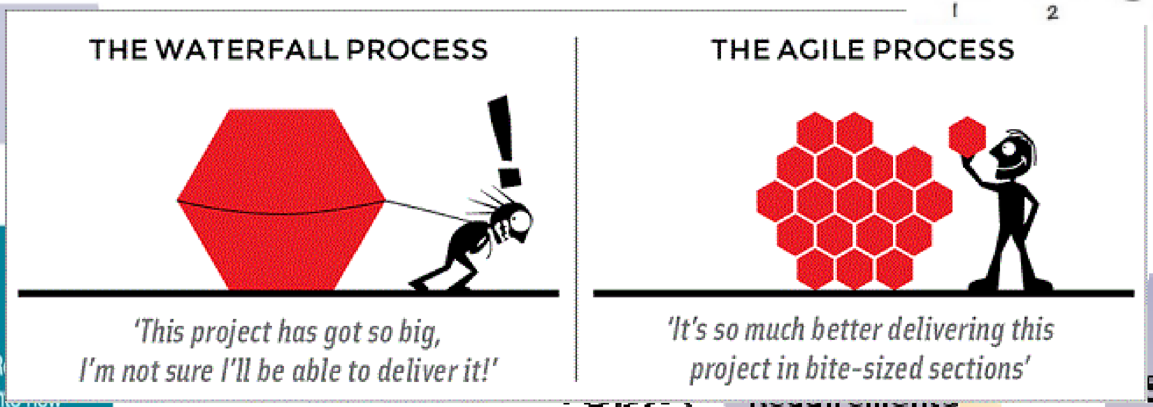
S/W Delivery Methods



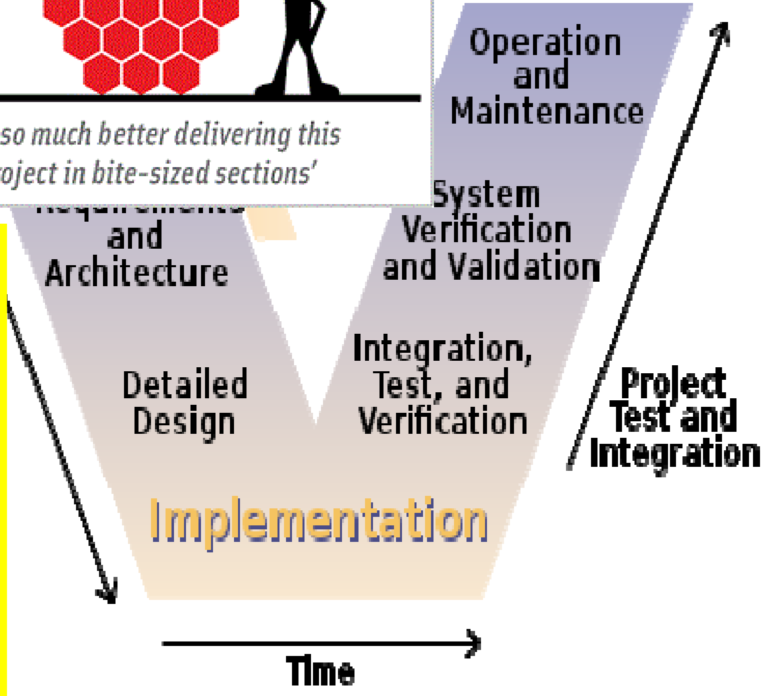


		←2 weeks→	←2–4 weeks→	←6–10 weeks→	
		design	build	gile prints and ustomer testing	
Create exceptional customer experience		Start zero-based design of digital process	Build prototype that delivers customer experience	Test with customers and refine	R
Transform business process	Set up end-to-end "SWAT team"	and scope minimum viable product	Detail and approve design of new business	Operationalize and scale business process	into new processes





THE WATERFALL PROCESS
'This project has got so big, I'm not sure I'll be able to deliver it!'
THE AGILE PROCESS
'It's so much better delivering this project in bite-sized sections'

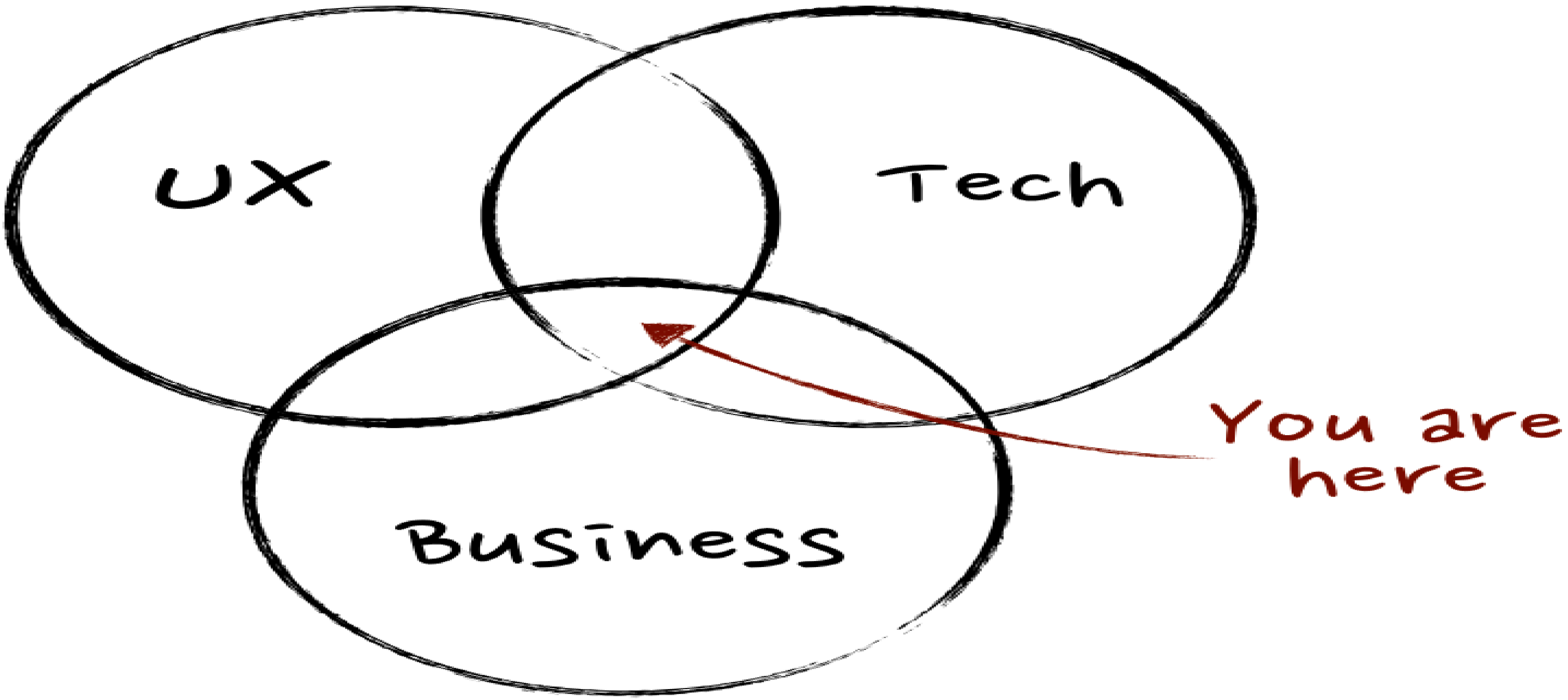


Requirements and Architecture
Detailed Design
Implementation
Integration, Test, and Verification
System Verification and Validation
Operation and Maintenance
Project Test and Integration
Time

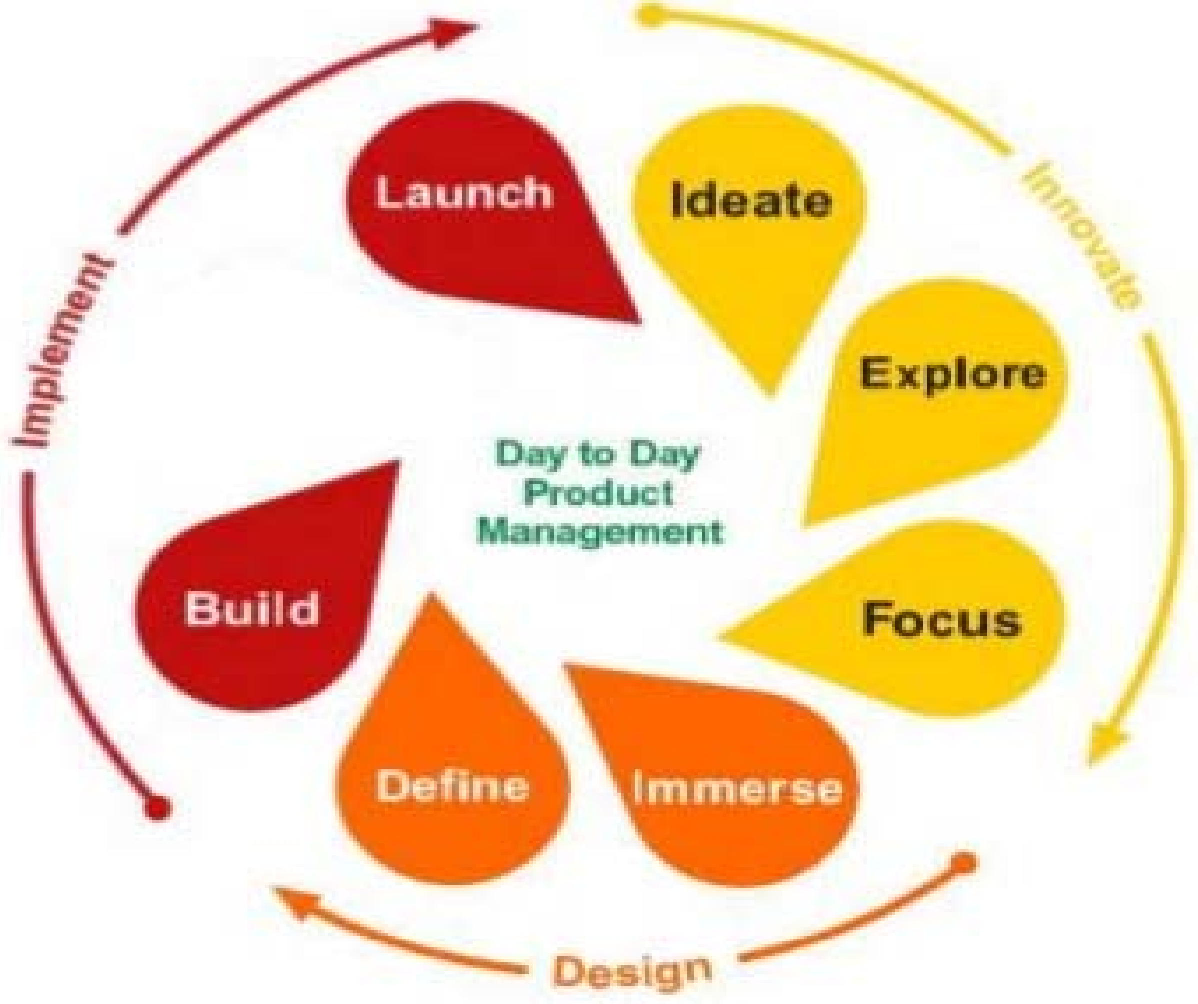
WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
I'M GLAD IT HAS A NAME.
THAT WAS YOUR TRAINING.

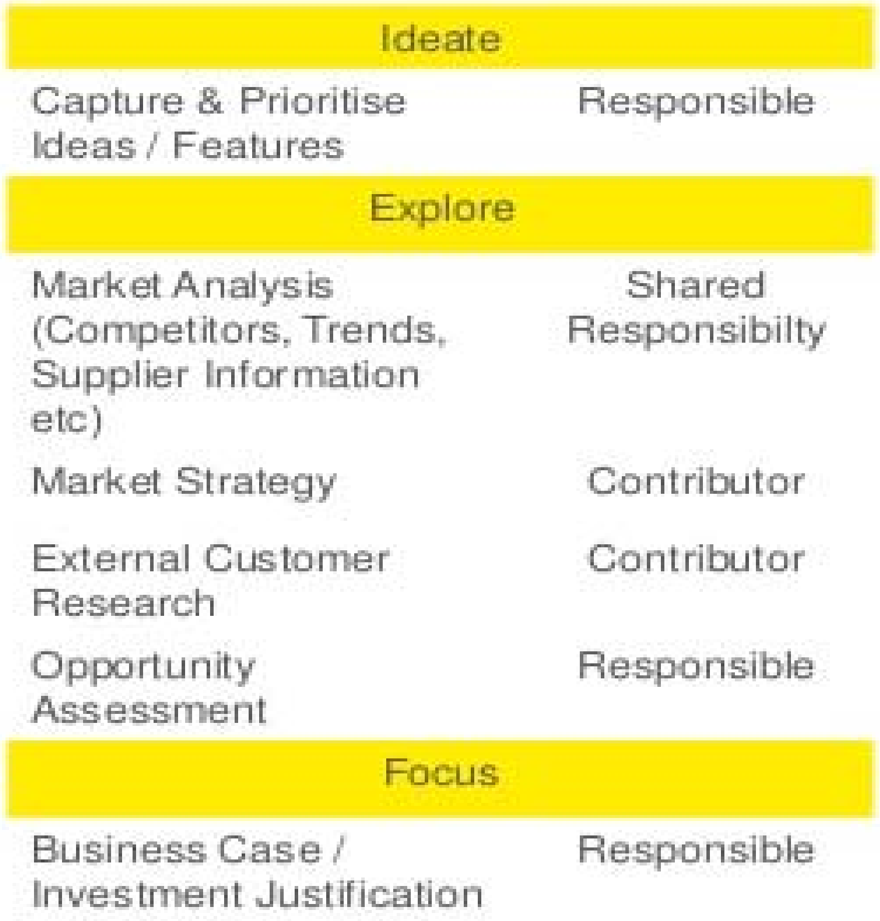
© Scott Adams, Inc./Dist. by UFS, Inc.

Product Manager



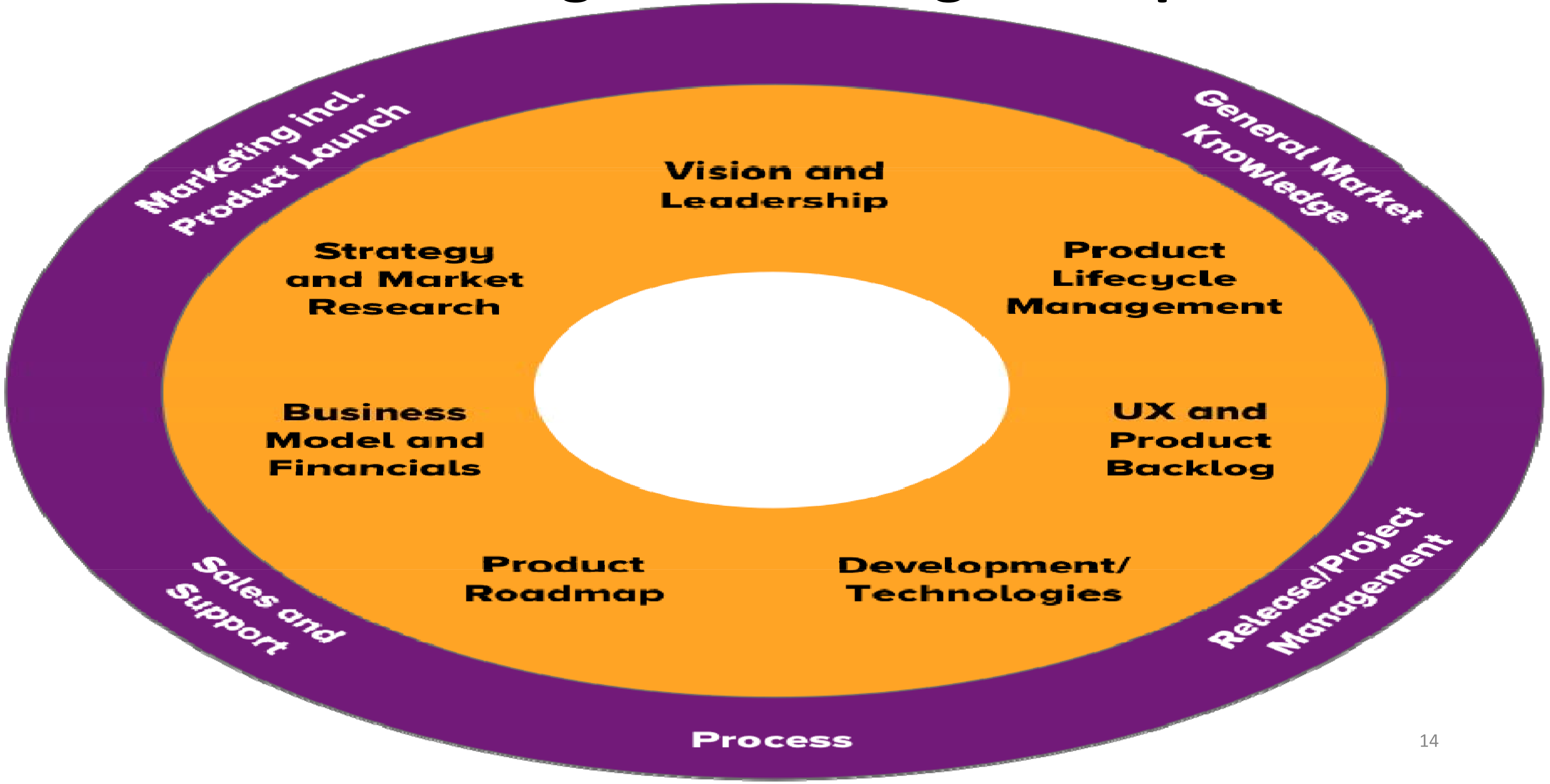
Product Manager Role



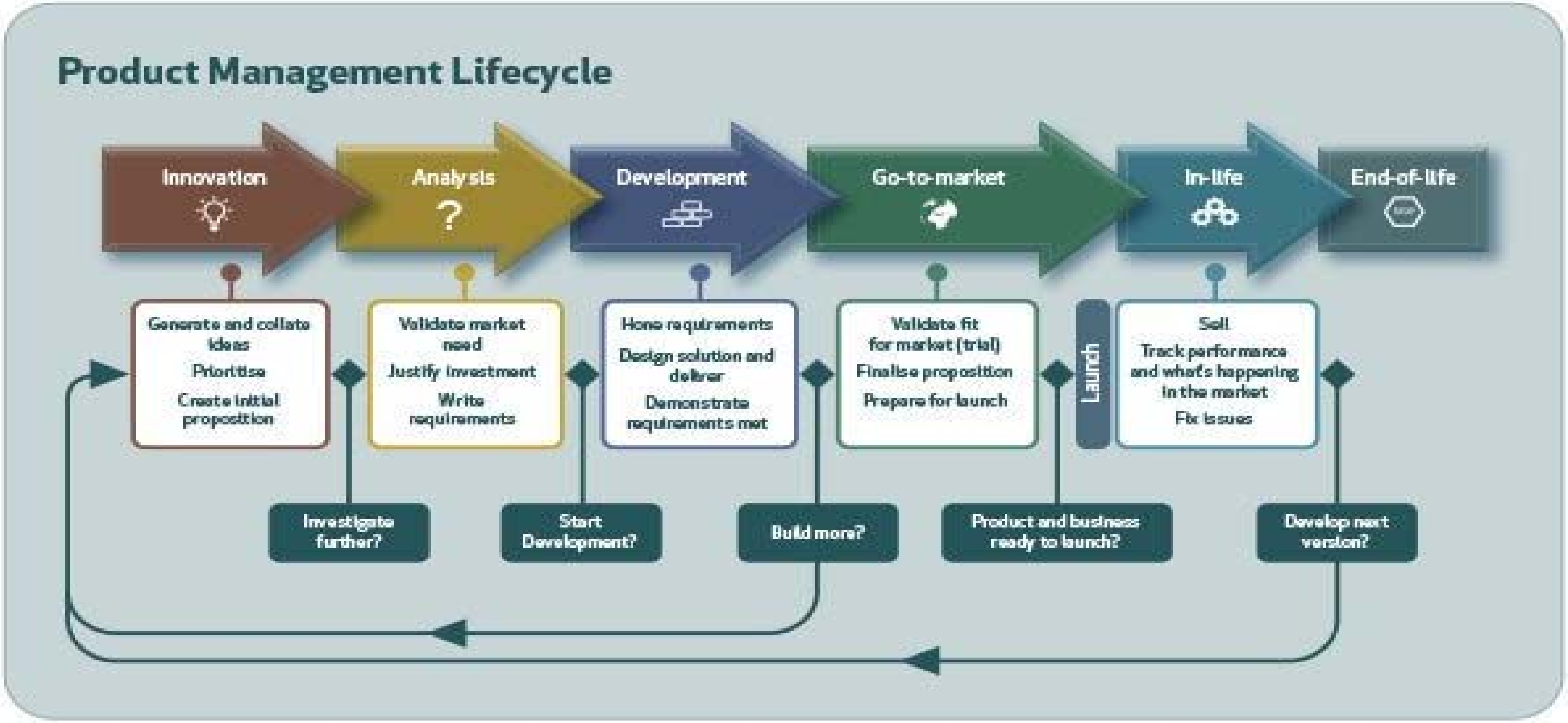


Ideate	
Capture & Prioritise Ideas / Features	Responsible
Explore	
Market Analysis (Competitors, Trends, Supplier Information etc)	Shared Responsibilty
Market Strategy	Contributor
External Customer Research	Contributor
Opportunity Assessment	Responsible
Focus	
Business Case / Investment Justification	Responsible

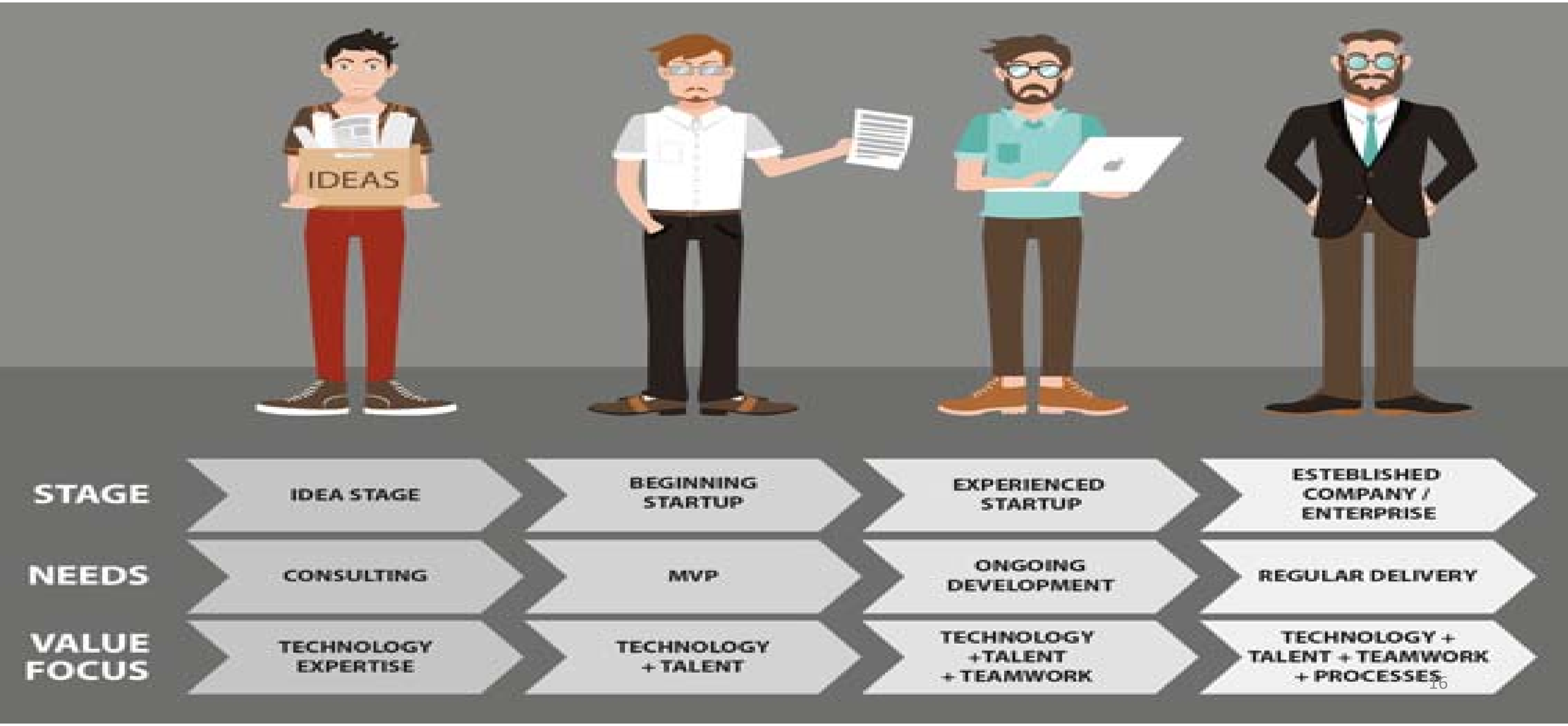
Product Manager Knowledge & Experience


Vision and Leadership
Strategy and Market Research
Product Lifecycle Management
Business Model and Financials
UX and Product Backlog
Product Roadmap
Development/ Technologies
Marketing incl. Product Launch
General Market Knowledge
Sales and Support
Release/Project Management
Process

Manage your Product Lifecycle



S/W company Stage



STAGE	IDEA STAGE	BEGINNING STARTUP	EXPERIENCED STARTUP	ESTEBLISHED COMPANY / ENTERPRISE
NEEDS	CONSULTING	MVP	ONGOING DEVELOPMENT	REGULAR DELIVERY
VALUE FOCUS	TECHNOLOGY EXPERTISE	TECHNOLOGY + TALENT	TECHNOLOGY +TALENT + TEAMWORK	TECHNOLOGY + TALENT + TEAMWORK + PROCESSES

Hippocampus.io

1 *We are an **innovation consultancy** re-shaping industries, ecosystems, and mindsets.*

2 *We work with the most forward thinking **Corporations**, **Organizations** and **Startups** in the Southern and Central Eastern Europe to find and build together the paths of disruptive innovation across industries and ecosystems.*

3 *We deliver innovation by embedding our startup mindset and culture into everything we do.*

4 *We have a unique mix of experience from both sides of the table as former corporate executives and entrepreneurs ourselves allowing us unique empathy with our corporate and startup clients and partners.*

Thank you

ΟΤΑΝ ΚΑΙΝΟΤΟΜΕΙΣ, ΠΡΕΠΕΙ ΝΑ ΕΙΣΑΙ ΠΡΟΕΤΟΙΜΑΣΜΕΝΟΣ ΟΤΙ ΘΑ ΣΕ ΠΟΥΝ ΤΡΕΛΟ.
LARRY ELLISON

Christos Lytras

Managing Partner @ Hippocampus.io

cl@hippocampus.io