UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION
CHAPTER 1 : EVOLUTION OF SOFTWARE HOUSE

LOO YIM LING
ylloo@utar.edu.my

Evolution of a software house – ACME Systems

- How ACME systems started
- The growing pains till successful
- Pros & Cons that follows
- The advantages with adopting CD and DevOps ways of working
- How it adapted to drive the business into new markets and opportunities

ACME Systems
- Humble Beginning

- Started out in the garage of one of the founders.
- The founders were visionaries with big ambitions, good ideas, and a little bit of cash.

ACME is purely fictional and based upon ACME Corporation,
first used in Road Runner Cartoons in the 1950s
– just in case you were wondering

ACME Systems
- Successful and Acquired

- After a few years of hard work, the dreams of the founders were realized.
- The business is recognized as a leader.
- It is acquired by an MNC.
- This acquisition brings in the funding & resources to scale the business
- However, with corporate owners comes corporate responsibilities, rules, bureaucracy, and processes.

ACME Systems
- Challenges to the Team

- The team starts to find it increasingly difficult and painful to deliver quality software.
- They adopt and adhere to the parent company's processes to improve quality and reduce risk, but this makes the seemingly simple task of delivering software, laborious and extremely complex.

ACME : How To React?
- ACME ver 1.0

- Typical start-up software business
  - Move fast to survive
  - Entice and retain customers at all costs.
- They do this by delivering what the customer wants just before the customer needs it, not too soon, not too late
- Various pre-sales prototypes needed to be built in a hurry
  - Mostly discarded
  - Some went to production

ACME : How To React?
- ACME ver 1.0: Characteristics

- Almost no barriers between developers and operations teams
- Developers have full access to the production environment and can closely monitor their software
- All areas of the business are focused on the same thing: to get software into the production environment ASAP
- Speed of delivery is of the essence

ACME : How To React?
- ACME ver 1.0: Characteristics

- When things break, everyone swarms around to help fix the problem
- The software evolves quickly and features are added in incremental chunks
- The ways of working are normally very agile

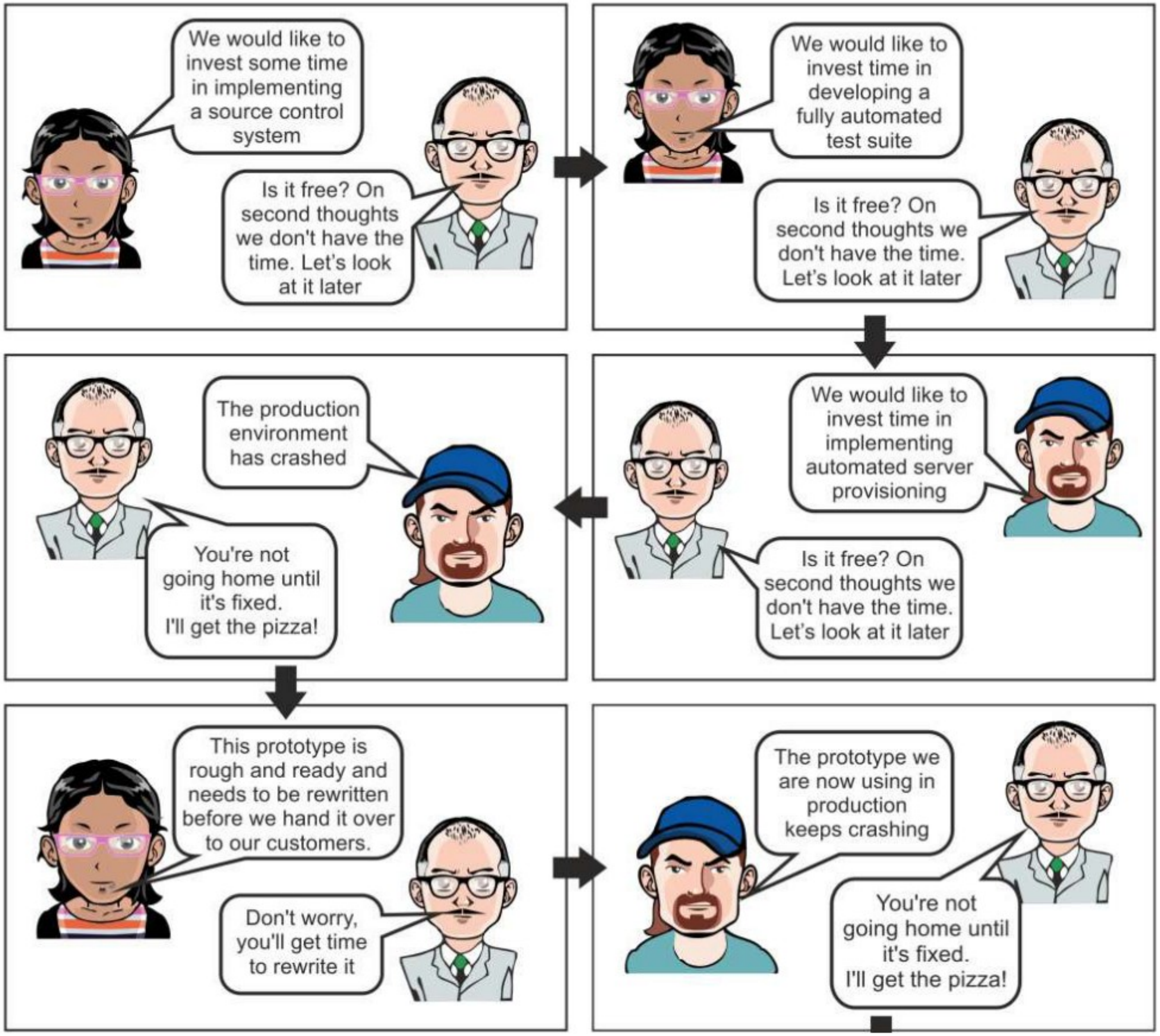
ACME : How To React?
- ACME ver 1.0: Possible Flaws

- Corners are cut to hit deadlines, which compromise:
  - software design and elegance
  - Application security
  - Engineering best practices
  - Robustness of software (with the lack of testing)
- Source and version control systems are **not used** religiously

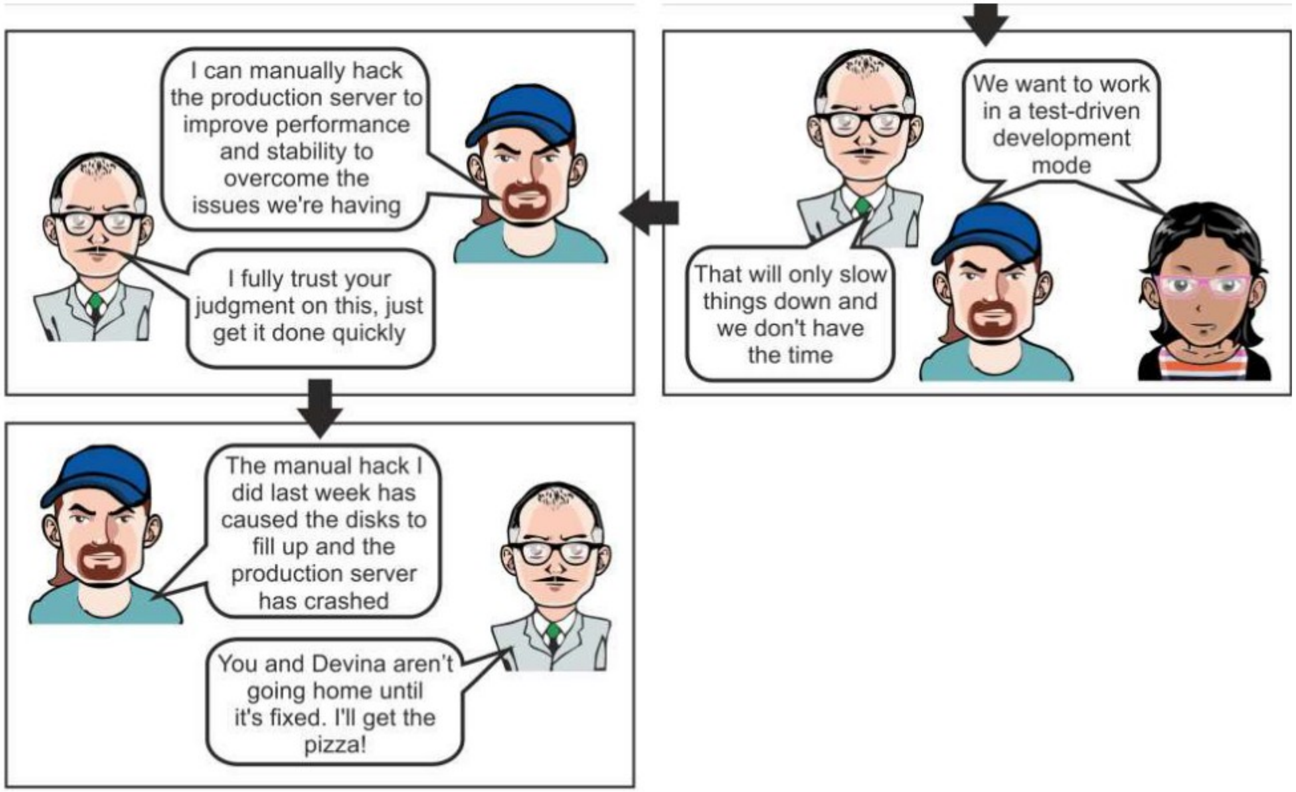
ACME : How To React?
- ACME ver 1.0: Possible Flaws

- Tweaks and changes can be made to the infrastructure with little or no audit trail
- Software releasing will be mainly manual
- At times, a rough and ready prototype may well become production code without the opportunity for refactoring
- Documentation is scant or non-existent
- The work-life balance for an engineer working within a small software house is not sustainable and burn out does happen

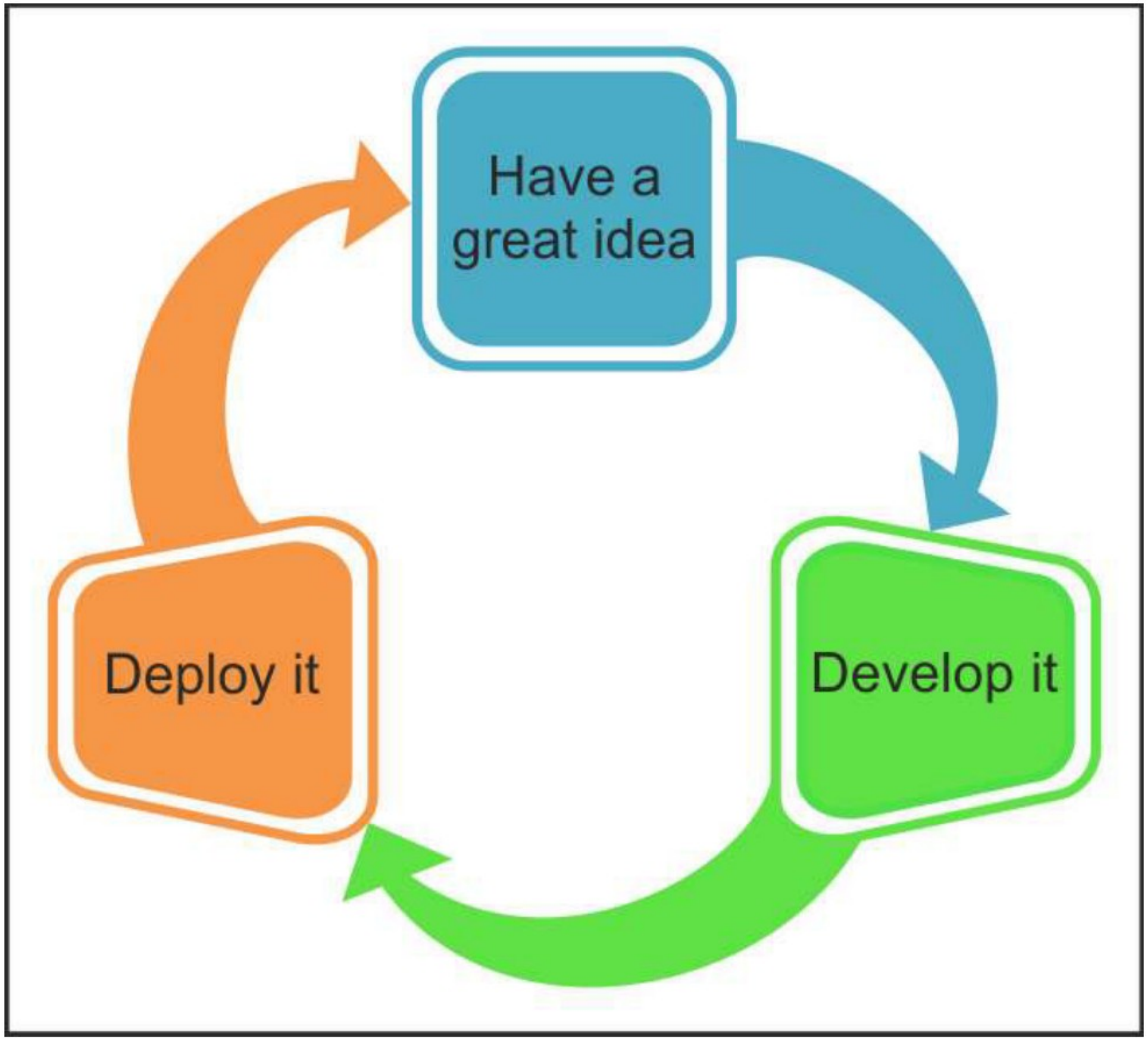
ACME : How To React?
- ACME ver 1.0


We would like to invest some time in implementing a source control system
Is it free? On second thoughts we don't have the time. Let's look at it later
We would like to invest time in developing a fully automated test suite
Is it free? On second thoughts we don't have the time. Let's look at it later
We would like to invest time in implementing automated server provisioning
Is it free? On second thoughts we don't have the time. Let's look at it later
The production environment has crashed
You're not going home until it's fixed. I'll get the pizza!
This prototype is rough and ready and needs to be rewritten before we hand it over to our customers.
Don't worry, you'll get time to rewrite it
The prototype we are now using in production keeps crashing
You're not going home until it's fixed. I'll get the pizza!

ACME : How To React?
- ACME ver 1.0


We want to work in a test-driven development mode
That will only slow things down and we don't have the time
I can manually hack the production server to improve performance and stability to overcome the issues we're having
I fully trust your judgment on this, just get it done quickly
The manual hack I did last week has caused the disks to fill up and the production server has crashed
You and Devina aren't going home until it's fixed. I'll get the pizza!

ACME : How To React?
- ACME ver 1.0: Software Delivery Proces Flow



ACME : How To React?
- ACME ver 2.0

- **Business has grown in size**
  - **Customer base is global**
  - **Software platform is used by millions**
  - **ACME well established, well renowned**
- **ACME is acquired by MNC**
  - **Sufficient funding to scale out and R&D**
  - **Admin, sales, marketing, etc can be passed to the parent company**

ACME : How To React?
- ACME ver 2.0

- R&D Team
  - introduces new tools and processes to enable speedy delivery of quality software.
  - Scrum is adopted across the R&D team and the opportunity to fully exploit engineering best practices is realized.
  - The original ACME systems platform starts to creak and is showing its age, so further investment is provided to re-architect and rewrite the software platform using the latest technologies.
  - In short, the R&D team feels that it's all starting to come together and they have the opportunity to do it right.

ACME : How To React?
- ACME ver 2.0

- Operations Team
  - absorbed as parent's global operations team.
  - data centers filled with cutting-edge kit, global network capabilities, and scalable infrastructure.
  - Everything that is needed to host and run the ACME systems platform is there.
  - Like the R&D team, the operations team has more than they could have dreamed of.
  - Operations team also has resources available to help maintain quality, control change to the platform, and ensure the platform is stable and available 24/7.

ACME : How To React?
- ACME ver 2.0

- Increasingly difficult to ship software
- Releases are getting more complex as the new platform grows with more features
- Despite re-architecting & rewriting, there still remains legacy code deep within the system
- Developers are far from the production environment, and are ignorant as to how the software performs, once it eventually goes live

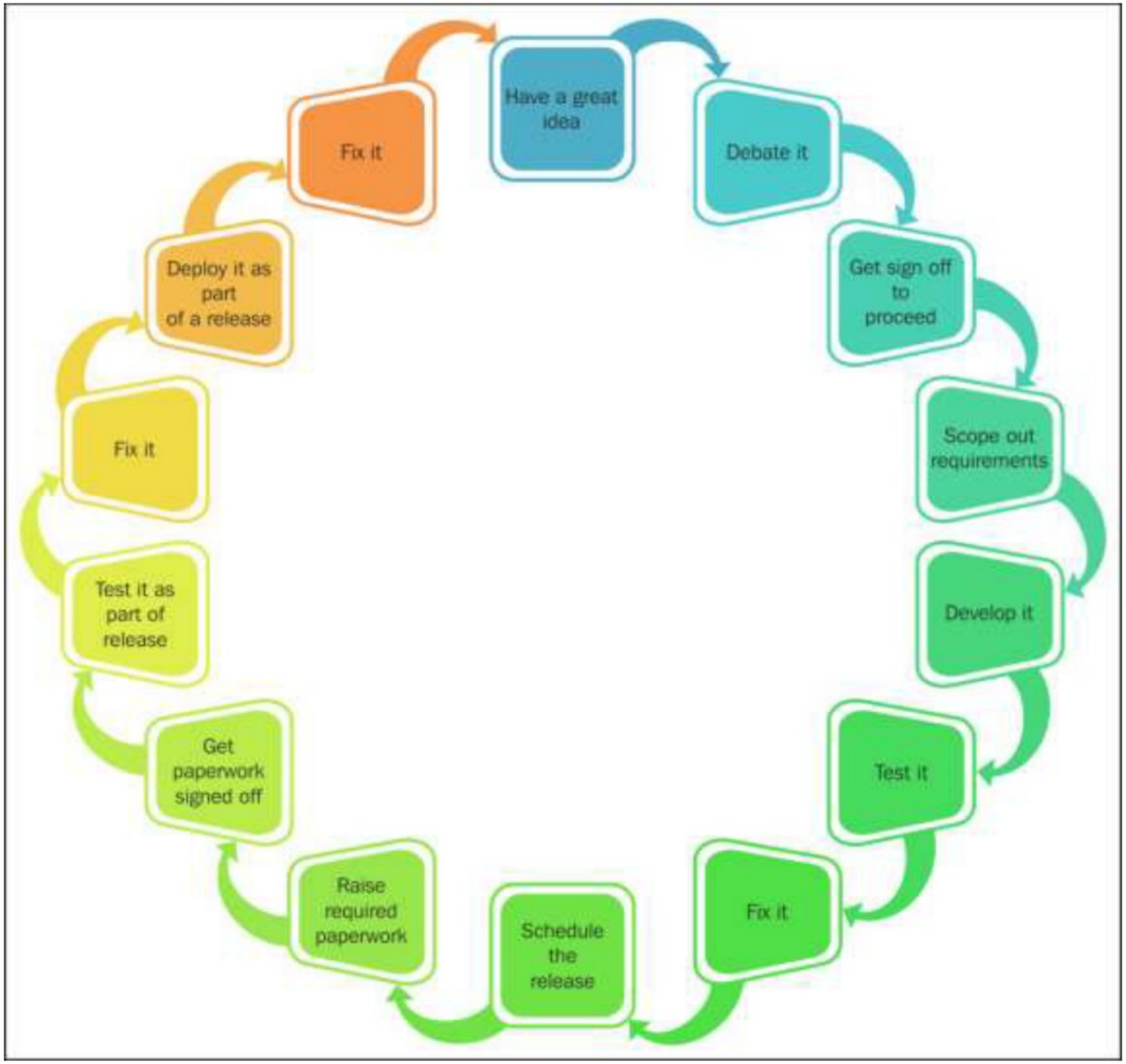
ACME : How To React?
- ACME ver 2.0

- There is a greater need to provide proof that software changes are of the highest quality and performance before they can go anywhere near the production servers
- Quality is starting to suffer as last minute changes and frantic bug fixes are being applied to fit into release cycles
- The technical debt amassed during the early days is starting to cause major issues
- Project scope is being cut at the last minute as features don't fit into the release cycles

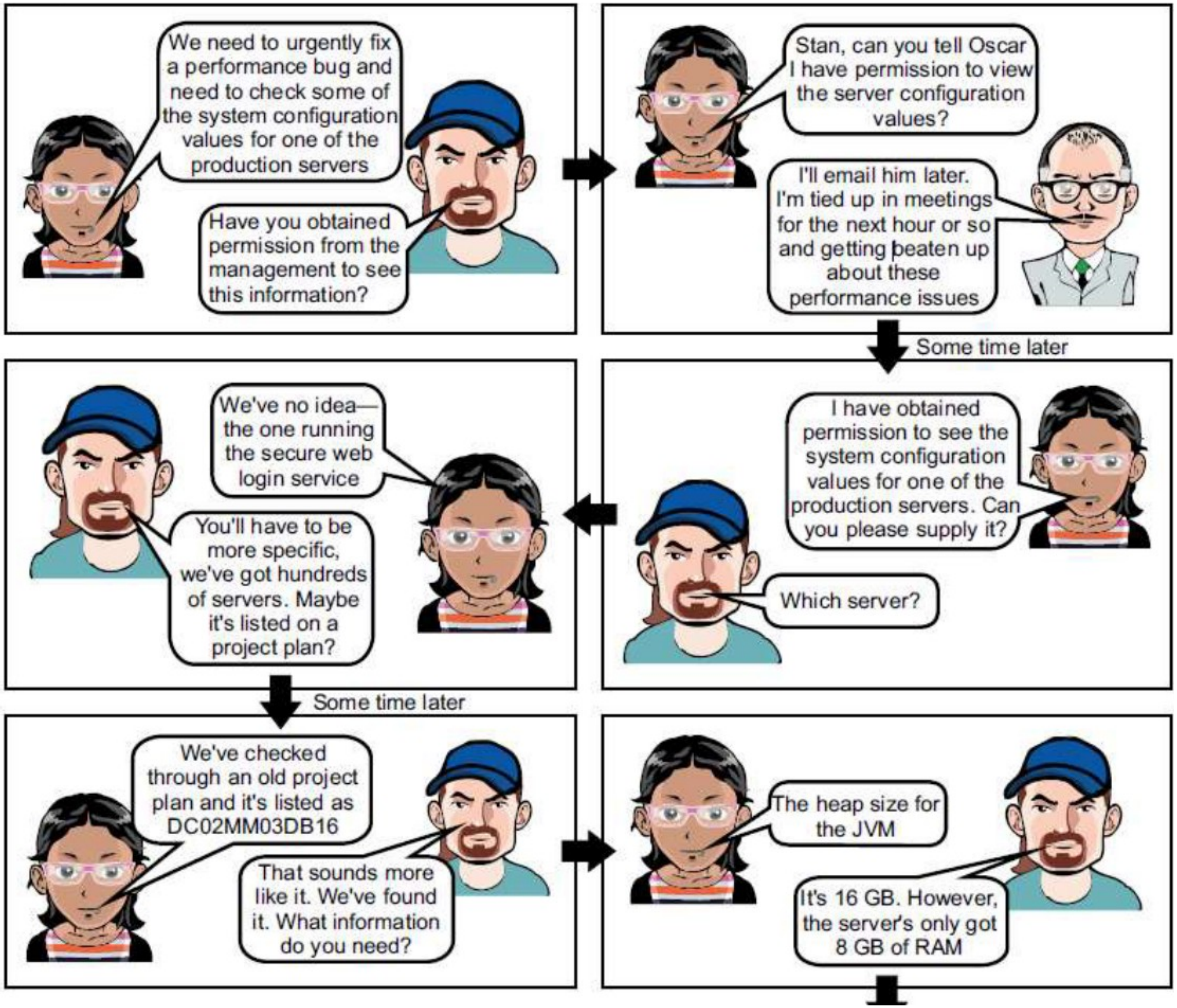
ACME : How To React?
- ACME ver 2.0

- More development resources are being applied to assisting releases
- Deployments are causing system downtime
- Deadlines are being missed, stakeholders are being let down, and trust is being eroded
- The business's once glowing reputation is being tarnished

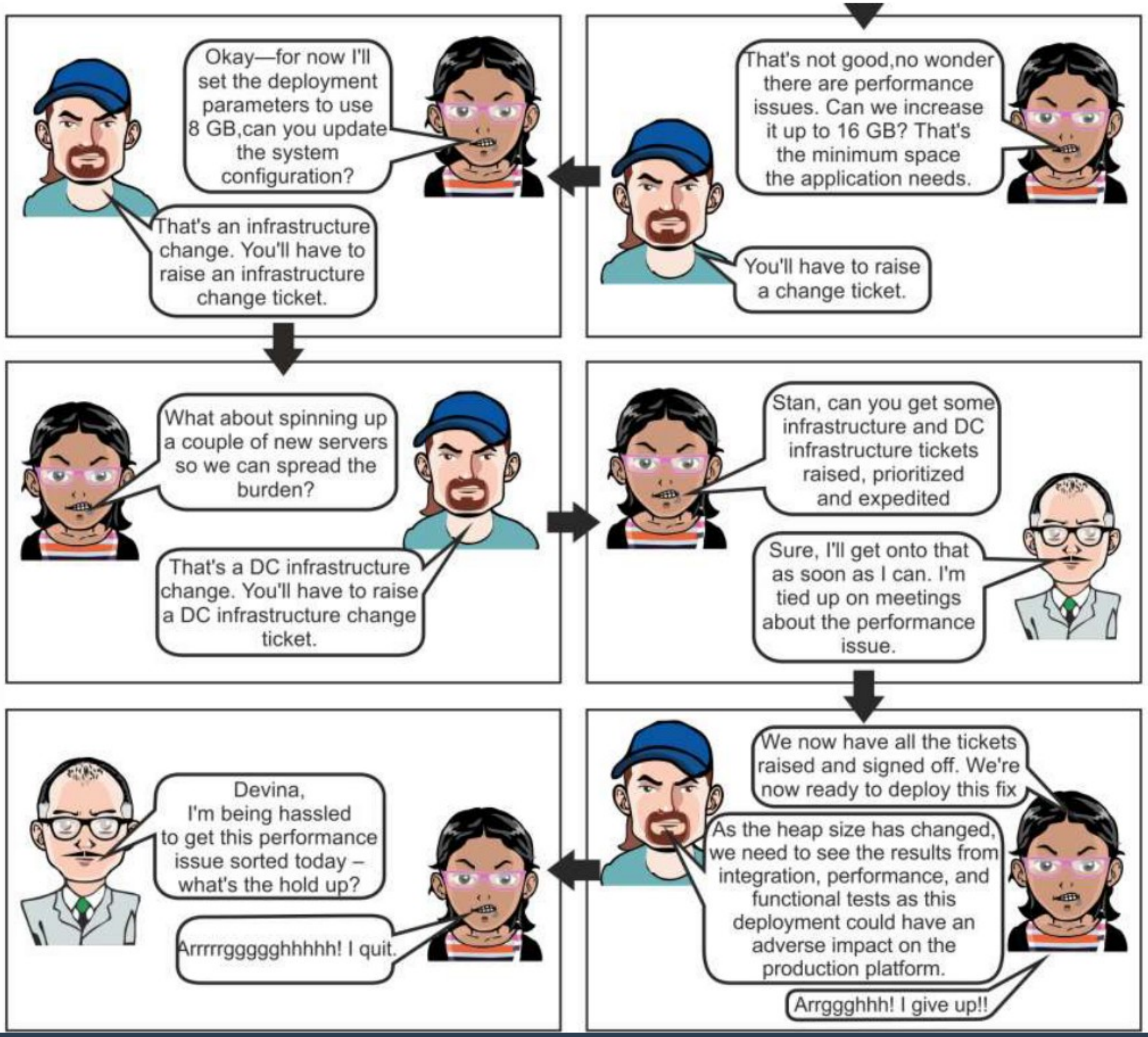
ACME : How To React?
- ACME ver 2.0: Software Delivery Process Flow



ACME : How To React?
- ACME ver 2.0


We need to urgently fix a performance bug and need to check some of the system configuration values for one of the production servers
Have you obtained permission from the management to see this information?
Stan, can you tell Oscar I have permission to view the server configuration values?
I'll email him later. I'm tied up in meetings for the next hour or so and getting beaten up about these performance issues
Some time later
I have obtained permission to see the system configuration values for one of the production servers. Can you please supply it?
Which server?
We've no idea— the one running the secure web login service
You'll have to be more specific, we've got hundreds of servers. Maybe it's listed on a project plan?
Some time later
We've checked through an old project plan and it's listed as DC02MM03DB16
That sounds more like it. We've found it. What information do you need?
The heap size for the JVM
It's 16 GB. However, the server's only got 8 GB of RAM

ACME : How To React?
- ACME ver 2.0


That's not good,no wonder there are performance issues. Can we increase it up to 16 GB? That's the minimum space the application needs.
You'll have to raise a change ticket.
Okay—for now I'll set the deployment parameters to use 8 GB,can you update the system configuration?
That's an infrastructure change. You'll have to raise an infrastructure change ticket.
What about spinning up a couple of new servers so we can spread the burden?
That's a DC infrastructure change. You'll have to raise a DC infrastructure change ticket.
Stan, can you get some infrastructure and DC infrastructure tickets raised, prioritized and expedited
Sure, I'll get onto that as soon as I can. I'm tied up on meetings about the performance issue.
We now have all the tickets raised and signed off. We're now ready to deploy this fix
As the heap size has changed, we need to see the results from integration, performance, and functional tests as this deployment could have an adverse impact on the production platform.
Arrggghhh! I give up!!
Devina, I'm being hassled to get this performance issue sorted today – what's the hold up?
Arrrrggggghhhhh! I quit.

ACME : How To React?
- ACME ver 2.0: Improving the Situation

- Small virtual team to identify the problems and fix the current situation:
    - Implementation of tools:
        - Build and test automation
        - Continuous Integration (CI)
        - Automated deployment and monitoring solutions
    - Change of organizational culture

ACME : How To React?
- ACME ver 3.0

- Formally known as CD Team
  - dedicated to addressing the problematic culture and behaviors, and developing ways to overcome and/or remove the barriers.
  - They are no longer simply a technical team; they are a catalyst for change.
  - Do whatever is needed to streamline the process of software delivery and make it seamless and repeatable.

ACME : How To React?
- ACME ver 3.0

- Run sessions to understand & map out end-to-end product delivery process
- Refining and simplifying tooling based upon continuous feedback
- Addressing the complexity of managing dependencies and order of deployment
- Engaging CD experts to assess the progress being made

ACME : How To React?
- ACME ver 3.0

- Arranging CD training and encourage both R&D and Ops team members to attend the training together
- Reducing the handover and decision-making points throughout the software release process
- Removing the barriers to allow R&D to safely deploy their own software to the production platform

ACME : How To React?
- ACME ver 3.0

- Working with other business functions to gain trust and help them to refine and streamline their processes
- Working with R&D and Ops teams to experiment with different agile methodologies such as Kanban
- Openly and transparently sharing information and data around deliveries and progress being made across all areas of the business

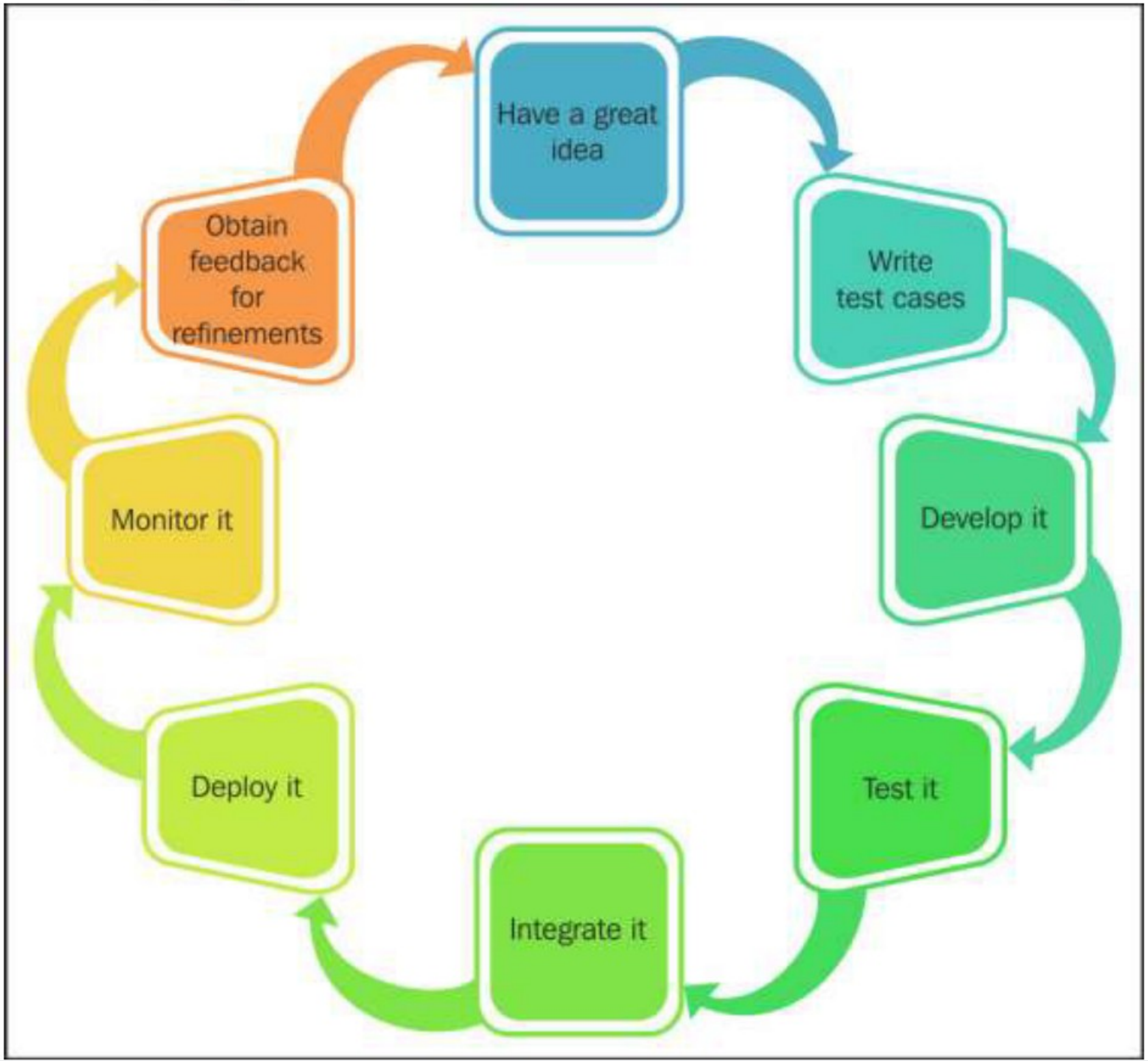
ACME : How To React?
- ACME ver 3.0

- Replacing the need for complex performance testing with the ability for developers to closely monitor their own software running in the production environment
- Evangelizing across all areas of the business to share and sell the overall vision and value of CD and DevOps

ACME : How To React?
- ACME ver 3.0

- **After some months, the process of building and delivering software has transformed to the extent that a code change can be built, fully tested, and deployed to the production platform in minutes, many times per day all at the press of a button and *initiated* and *monitored* by the developer who *made the change*.**

ACME : How To React?
- ACME ver 3.0: Software Delivery Process Flow



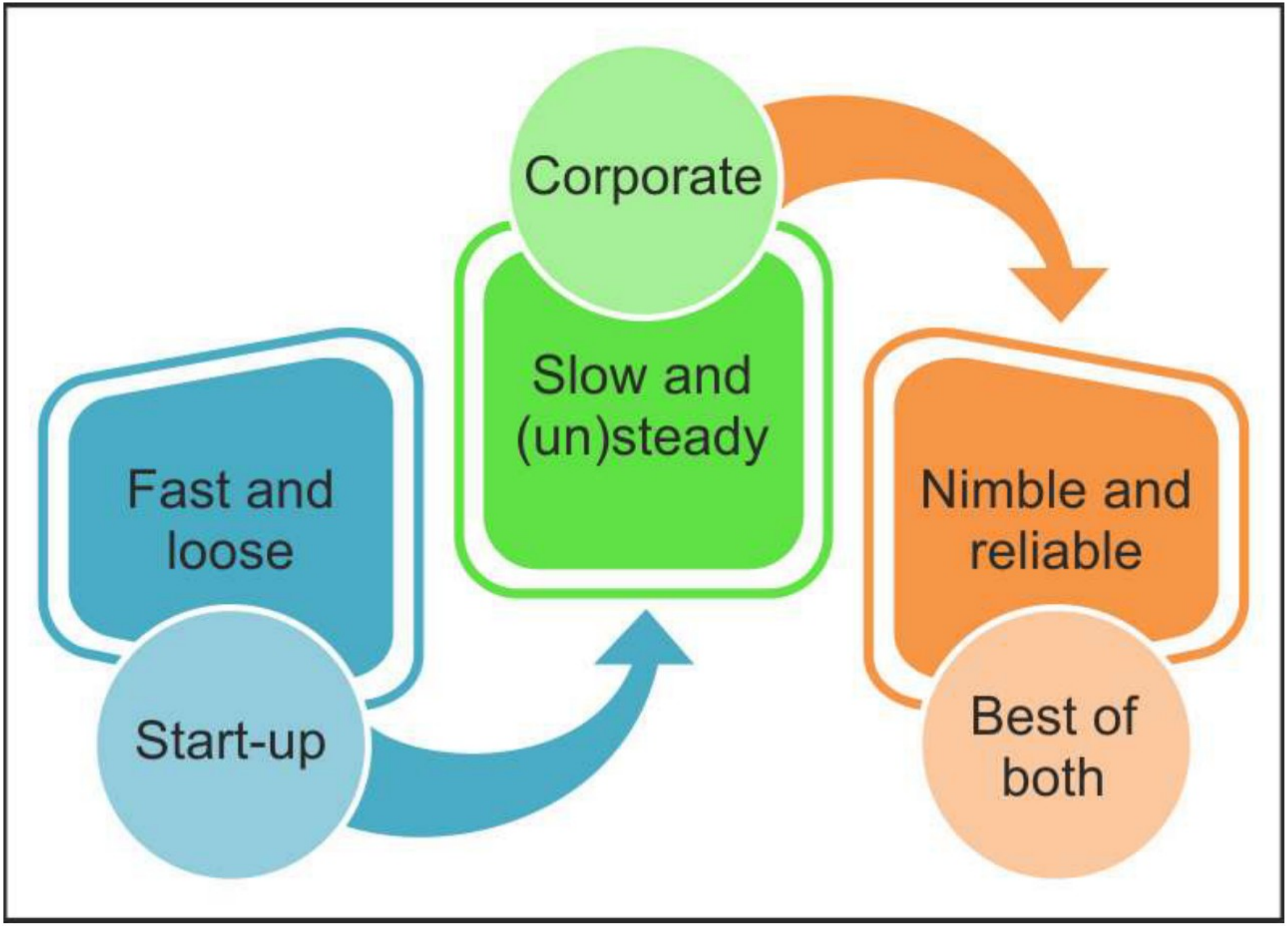
ACME : How To React?
- ACME ver 4.0

- New technical challenges
  - From web to mobile
  - From own data centers to cloud solutions
  - …

ACME : How To React?
- ACME ver 4.0

- Actions
  - Agree on a realistic delivery cadence to allow for regular incremental changes without bombarding the end consumer
  - Invest in new automated build, CI, and testing tools, which seamlessly integrate with and enhance the existing tooling
  - Invest in nonfunctional features that will allow for greater visibility of what is running out in the wild, which again seamlessly integrates with the existing tooling and monitoring approach
  - Ensure that the engineers delivering the mobile clients work closely with the backend engineers (DevOps) so that the client integrates seamlessly and doesn't cripple the existing production platform

Evolution in A Nutshell



END OF LECTURE 02