Keynote

Current Trends in Software Engineering and the Implications of their Convergence

Roy Oberhauser
Aalen University
Germany
“IT operations is challenged by the rapid growth in data volumes generated by IT infrastructure and applications that must be captured, analyzed and acted on”
- Padraig Byrne, Senior Director Analyst at Gartner
A crazy IT deployment landscape

Amazon Microservices

Netflix Microservices

The complex IT reality today

57% of enterprises
use between
1000-5000
business applications
40% of IT Staff will be versatilists by 2021, holding multiple roles, most of which will be business-related rather than tech-related.

- Gartner
83% of IT organizations enable/support self-service tools for end users [BMC]

Tier 0
Self-service, Crowd-sourcing

Tier 1
Basic (record, respond)

Tier 2
In-depth

Tier 3
Subject-matter expert

Tier 4
Outsourced

“Left-shifting” IT support
AI and Virtual Agents to the rescue

ITSM (Service Management) Gartner Hype Cycle (2020)
Towards NoOps: Virtual Support Agents

25% of customer service and support Ops will integrate virtual customer assistant (VCA) or chatbot tech by 2020 across engagement channels, up from <2% in 2017.

- Gartner
NoOps via AIOps: Artificial Intelligence

- 40% of large enterprises using AIOps by 2022 to support and partially replace IT Ops activities (up from 5% today)
- Large enterprise exclusive use of AIOps and digital experience monitoring tools will increase from 5% (2018) to 30% in 2023
- AIOps will have a long-term transformative impact on IT operations

- Padraig Byrne, Senior Director Analyst at Gartner

Towards NoOps: Serverless

Over 50% of AWS users are using the serverless AWS Lambda Function as a Service (FaaS)

© 2020 Roy Oberhauser
Towards NoOps: Serverless & Intelligent Infrastructure

Gartner Hype Cycle Infrastructure Strategies (2020) - Selection

[Adapted from Gartner]
Towards NoOps: Infrastructure Automation

Agile and DevOps Gartner Hype Cycle (2020) – Selection
DevOps: What are the implications?

ITSM (Service Management) Gartner Hype Cycle (2020)

© 2020 Roy Oberhauser
My entire company has fully embraced DevOps

The majority of teams within my company are practicing DevOps

A few teams within my company are fully immersed in DevOps

We have just undertaken our DevOps journey

We are considering DevOps, but have not yet started

We have not adopted DevOps and have no plans to adopt

I’ve never heard of DevOps before

Note: Worldwide; 2016 to 2018; 1091 Respondents; technology professionals responsible for development and quality of web and mobile applications

Source(s): Sauce Labs, Dimensional Research
DevOps: 5 Stage Evolution Model

Stage 1: Normalization
- Application development teams use version control
- Teams deploy on a standard set of operating systems

Stage 2: Standardization
- Teams deploy on a single standard operating system
- Build on a standard set of technology

Stage 3: Expansion
- Individuals can do work without manual approval outside team
- Deployment patterns for building apps/services are reused
- Infrastructure changes are tested before deploying to production

Stage 4: Automated infrastructure delivery
- System configurations are automated
- Provisioning is automated
- System configs are in version control
- Infrastructure teams use version control
- Application configs are in version control
- Security policy configs are automated

Stage 5: Self-service
- Incident responses are automated
- Resources are available via self-service
- Applications are rearchitected based on business needs
- Security teams are involved in technology design and development

Graph showing percentage of respondents in different stages of evolution:
- High evolution: 11%
- Medium evolution: 75%
- Low evolution: 10%

© 2020 Roy Oberhauser
## DevOps: Potential

<table>
<thead>
<tr>
<th>High performing DevOps organizations</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>46X more code deployments</td>
<td></td>
</tr>
<tr>
<td>440X faster commit ➔ deploy</td>
<td></td>
</tr>
<tr>
<td>96X faster recovery from downtime</td>
<td></td>
</tr>
<tr>
<td>Changes 20% less likely to fail</td>
<td></td>
</tr>
<tr>
<td>21% less time on unplanned (re)work</td>
<td></td>
</tr>
<tr>
<td>44% more time on new work</td>
<td></td>
</tr>
</tbody>
</table>

© 2020 Roy Oberhauser
DevOps: Build Deployment Frequency (2018)

Desired frequency vs. Typical frequency for deployment frequency:
- Hourly: Desired 16%, Typical 5%
- Daily: Desired 33%, Typical 27%
- Weekly: Desired 22%, Typical 24%
- Less often than weekly: Desired 28%, Typical 44%

Source: Sauce Labs; Dimensional Research, 2018
136,000 deployments per day.

Ken Exner
Amazon Director of Development Productivity
Deployment frequency has come a long way since 2012

Application Development and Delivery Gartner Hype Cycle (2018)

#Post-Scrum Methodologies
# Integrate Anything: x-PaaS *(Platform-as-a-Service)*

<table>
<thead>
<tr>
<th>Application Dev</th>
<th>Platform Services</th>
<th>(adPaaS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Platform Services</td>
<td>(aPaaS)</td>
</tr>
<tr>
<td>High-productivity app</td>
<td>Platform Services</td>
<td>(hpaPaaS)</td>
</tr>
<tr>
<td>Business Analytics</td>
<td>Platform Services</td>
<td>(baPaaS)</td>
</tr>
<tr>
<td>Business Process</td>
<td>Management Services</td>
<td>(bpmPaaS)</td>
</tr>
<tr>
<td>Business Rule</td>
<td>Platform Services</td>
<td>(brPaaS)</td>
</tr>
<tr>
<td>Citizen Integrator</td>
<td>Software as a Service</td>
<td>(iSaaS)</td>
</tr>
<tr>
<td>Integration</td>
<td>Platform Services</td>
<td>(iPaaS)</td>
</tr>
<tr>
<td>Communications</td>
<td>Platform Services</td>
<td>(cPaaS)</td>
</tr>
<tr>
<td>Database</td>
<td>Platform Services</td>
<td>(dbPaaS)</td>
</tr>
<tr>
<td>Data Broker</td>
<td>Platform Services</td>
<td>(dbrPaaS)</td>
</tr>
<tr>
<td>Event-Processing</td>
<td>Platform Services</td>
<td>(epPaaS)</td>
</tr>
<tr>
<td>Function</td>
<td>Platform Services</td>
<td>(fPaaS)</td>
</tr>
</tbody>
</table>

[Gartner: Platform as a Service: Definition, Taxonomy and Vendor Landscape, 2016]

© 2020 Roy Oberhauser
Towards Complete Software Infrastructure

Application Development and Delivery

Gartner Hype Cycle (2018) - Selection

In the API Economy with Public Web APIs, everything becomes programmable.
Towards Software-Defined Architecture (SDA)

- SDA creates a virtualization layer over the software APIs by providing an API gateway.
- Enables reconfiguring the software application infrastructure without necessarily needing a software architect or developers.
- As the availability of Public Web APIs increase, SDA will enable applications to be quickly reconfigured or redefined based on changing circumstances.
Towards AI-empowered Developers: Augmented SW Development and the Virtual Developer

At least 40% of new Application Development projects will have an AI-powered ‘virtual developer’ on their team by 2022.

- Gartner (2018)
Towards Autonomous Testing

- Leverages AI & ML to reduce the amount of manual labor associated with creating test code
- Enables Citizen Developers to have their code tested without necessarily requiring additional develop/tester resources
Democratization of Software Development

The rise of the Citizen Developer in a tech-savvy population

“We’re all developers now”
- Gartner (2012)
Low-code platforms will be used in more than 65% of software development projects by 2024
- Gartner

Application Development and Delivery
Gartner Hype Cycle (2018) - Selection

#Citizen Developers
NoDevs?: Left-Shifting Engagement Trend

Now

BizUser | BizAnalyst | Developer | IT

Future?

BizUser | BizAnalyst | Developer | IT

© 2020 Roy Oberhauser
Towards NoDevhs: AI may eventually replace (some subset of) developers

- Various ongoing or completed projects are addressing AI-based code generation:
  - BAYOU
  - Deep TabNine – autocompleter trained on 2M GitHub repos
  - OpenAI code generation example via GPT2

- By 2040, ML and NLP technologies are expected to be capable of writing better software code faster than the best human developers. [Billings et al.: "Will humans even write code in 2040 and what would that mean for extreme heterogeneity in computing?", 2017]
#NoCode
At least currently serious challenges remain for AI/ML assisted SW development

- We still have a way to go, even for automatically detecting known software design patterns.
  - E.g., refer to the ICSEA 2020 paper:
    - “A Machine Learning Approach Towards Automatic Software Design Pattern Recognition Across Multiple Programming Languages”

- We still face various challenges for creating intelligent autonomic process-aware information systems – refer to book
Keep hoping for No Silver Bullet? Will Software’s Essential Difficulties Remain?

Or can/will Augmented/Virtual Developers and No-Code Platforms do away with the essential difficulties of software development as elucidated by Frederick P. Brooks, Jr.:

- Complexity
- Conformity
- Changeability
- Invisibility

Finding A Value Stream for Software Developers
“The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later. Therefore, the most important function that the software builder performs for the client is the iterative extraction and refinement of the product requirements.”

-- Frederick P. Brooks, Jr.
Perhaps identifying stakeholders and getting stakeholder agreement and buy-in on explicit and implicit requirements, needs, and expectations may be the most challenging problem for AI to address.
Thank you!