



Why DevOps, TestOps and TEM? buzzwords or not?.

So NGSoft Inc, the fortune 500 company with a net income of 50 billion dollars based in Germany, outsourced their software development work to a third party company based in India and testing was also outsourced to another company based in Brazil.

The CTO, Dr Tayden at the last C-level meeting said "By outsourcing our development and testing work to third party companies, we can focus on our core business and reduce our investment in IT, year on year by 10% over the next 5 years."

The company's main data centre, operations and IT department were all based in Germany, so to successfully deliver a stable application or to upgrade to the latest release of their products trading software application would require that the in house IT team (or rather what is left of it after downsizing as implied in Dr Tayden's quote) and the third party companies work together seamlessly, in a highly organised and structured manner to successfully deliver the new release.

The new version will provide a very significant ROI, with an estimated net income increase of over 30% across the next three years, because the new version has built in CRM, a marketing campaign manager and captures new product lines.

All sound too familiar?....., and so all through the chain up to C-Level management, the pressure is on to deliver against all odds, as was also emphasised by the CEO at the last company social gathering four months ago at the five star Hilton hotel in Berlin.

The experts tell us that DevOps is a combination of;

Firstly, best practise framework incorporating elements of the existing best practise guidelines, ISO Standards and development methodologies to include Prince 2, ITIL, Scrum, Waterfall and Traditional delivery methods etc.

Secondly, automation and integration ensuring that from requirements gathering through to the application "*going live*" and now in "*operations*" particularly for a scenario such as the above mentioned, the delivery approach includes out sourcing, different teams in dispersed locations working together, different company ethos or work model, different IT networks, different IT Environments, restricted firewall access etc, which is a possible recipe for disaster...

Thirdly and probably most controversially, that a role called DevOps Engineer or Consultant or Architect etc will exist and be skilled to operate as an expert across the end to end cycle from requirements gathering through to supporting a live production system.

This role apparently will serve to provide the communication/collaboration linkage that brings together seamlessly, Development, Testing and Operations using all the above mentioned tools and techniques etc to ensure that the new version is successfully implemented and is stable.

TestOps is also now emerging and again the experts have borrowed from DevOps in defining it, obviously starting from the point of testing (instead of development) through to operation.

Let us delve in a little further then, shall we? so TestOps incorporates existing best practise in testing, adhering to testing standards, framework, also incorporates tools (cloud based or physically hosted), techniques and practise.

The end to end automation/integration of the testing execution cycle, collation of results, test preparation, promotion of successfully tested code to production is apparently all part of TestOps. Again a new role profile is emerging similar to DevOps and is called a TestOps Engineer.

So following all the above mentioned; for IT Environments, the management of it, automating its provisioning, code deployment, configuration management, tracking/monitoring of settings/configurable item or parameters, managing the scheduling, booking out through automated methods, process and strategy essentially it seems, all exist to underpin both DevOps and TestOps as the way forward in successfully delivering, fit for purpose/stable applications, in a cost effective manner.

In the end, the questions that come to mind are as follows;

- Are these new buzzwords here to stay or are they simply a regurgitation of existing practise, methods, process and obviously the natural progression of automation and integration which can be argued is simply an effect of Information technology advancement?

- Should C-Level management be concerned? is there a need for a knee jerk reaction to suddenly allocate a budget for these fairly new acronyms, create a new team, recruit DevOps and TestOps engineers and follow the trend etc?

- Is it possible that DevOps and TestOps should rather be seen as a guide or a measure ensuring that the right level of attention is paid to automation and integration of process, best practise, tools from requirements gathering through to running a system in its production state?

Please comment and assist in answering the questions raised.