ELECTRIC DREAMS
30th ascilite Conference, 1-4 December 2013
Macquarie University, Sydney

The Introduction of an Advanced Class in Systems Administration at Otago Polytechnic

Tom Clark
College of Enterprise and Development
Otago Polytechnic

Systems administration is a common career path for tertiary computing students, but it is difficult to take classes in the topic, especially at more advanced levels. Most of the classes that are available focus on specific tools and practices, often tied to particular vendors' systems. A set of topics around which to build a systems administration curriculum has not been clearly identified. At Otago Polytechnic we have developed a class that builds the specific knowledge and skills required to produce work-ready Systems Administrators.. The staff organised the class around a simulated workplace model rather than a more traditional lecture/lab model. This model emphasises having students perform tasks that are, as nearly as possible, identical to the tasks that they will eventually perform in a workplace. While the first instance of the class was generally successful, some issues, especially with assessment, were noted.

Keywords: Systems administration, Operations, Education

Rationale for the Class

Systems administration is the practice of installing, configuring, maintaining, troubleshooting, and generally managing computer and network systems (Nemeth, Snyder, Seebas, & Hein, 2001). This role emerged, at least in its present form, after about 1970. Since that time the role has evolved considerably and it continues to evolve at a rapid pace. (Most notably, the role was originally referred to as *System Administration*, indicating that a single server was involved. Now that even small organisations typically operate multiple servers, we typically refer to it as *Systems Administration*.) We might expect a job that has existed for over 40 years to be very well defined. However, this is not the case for systems administration. Even the League of Professional Systems Administrators says, ``... we're still not a profession," in part because the field has not identified a body of common wisdom nor the people to uphold and teach it. (LOPSA)

Because the systems administrator's role is hard to define, it is also hard to obtain training to become a systems administrator. It is possible to receive training in certain systems administration tasks, for example through training and certification programs offered by vendors like Cisco (Cisco Systems), Microsoft (Microsoft) and Red Hat (Red Hat), and these programs also inform the curricula of tertiary education programs. But these courses focus on basic skills without providing an opportunity for students to synthesise those skills into a body of practice. Whereas a student learning software engineering starts with elementary programming classes and progresses to increasingly advanced topics, a student learning systems administration is often limited to taking isolated and sometimes only elementary classes. The problem has a chicken-and-egg quality to it: it is hard to design a class in systems administration because the nature of the field isn't clearly defined, but it is also hard to describe a profession for which there is little training available.

Otago Polytechnic decided to address this problem by offering an advanced class in Systems Administration beginning in 2013. The staff were convinced that it would be possible to identify a collection of high level topics and tasks that would be relevant to aspiring systems administrators and that were not covered in other classes in the degree. This view was supported by discussions and emails with working systems administrators. In practice the challenge was not finding enough material, but rather narrowing the list of topics into one that could be reasonably covered in a single semester. The topics chosen include time and workflow management, documentation methods, centralised monitoring and configuration management, and communication skills.

Approach

The class was not organised around a collection of lectures and labs. Instead, it was designed to provide a set of connected experiences that allowed students to build and apply the same skills they would eventually use in their workplaces. This was accomplished using a simulated workplace class model that replicated as realistically as possible the actual experience of being a systems administrator. The model was an implementation of Work Integrated Learning, an approach that has been shown to build professional capabilities in students (Billett, 2011). Students worked in small teams to build, document, and operate a multi-tier web application. They used industry standard tools to monitor, manage, and back up their systems. They were assessed on the basis of how well their systems worked and on the quality and professionalism of their work practices. The goal was that students learn how to be systems administrators rather than learn some material that systems administrators happen to know.

To ensure that the class accurately modeled industry practices, working systems administrators were consulted for recommendations on the tasks and tools used. Many suggested multi-tier web applications as examples of realistic tasks with the desired level of complexity. One commented, ``A multi-node load balanced web frontend with some middleware and then a clustered DB setup. I think having seen data flow through a system and being able to hold all that in your head is often missing. Troubleshooting lots of moving parts will teach them how to exclude what isn't the problem." (Powell, 2012)

The course was built upon three conceptual pillars:

Real Tasks

Students should have the opportunity to work on systems that are identical to those that are used in the industry. This can be accomplished by directing the students to install, configure, and operate a real service that is deployed in the field. Just as a child is taught to ride a bicycle by giving her a bike and helping her attempt to ride it, a student may be taught to carry out professional tasks by giving her actual tasks and helping her attempt to perform them.

Real Tools

As the students are working on real tasks, they should perform those tasks using the same tools they will use after graduation. Most of the required tools are software that can be easily supplied to students. But the tools may also include online services or network infrastructure, and course preparation may include setting up those resources.

Real Assessment

When students are working on real tasks it makes sense to assess their work according to real, professional standards. Students can be evaluated on whether their services work reliably, whether they perform well, whether their systems are well documented, whether the students successfully solve problems that arise, and whether they can clearly and accurately describe their work.

Tasks

The class was organised around a large primary task that would occupy the entire semester: installing, configuring, deploying and operating a multi-tier web application. A web application deployment was chosen because it required setting up multiple systems, because it provided a good platform to address issues of performance, scaling, and security, and because it was fairly easy to simulate the actions of large numbers of users using commonly available web testing tools. But most importantly, the task was chosen because it is exactly the sort of task performed by working systems administrators. One large task such as this helped create a more realistic experience than a set of disconnected exercises would. This main task naturally divided into discrete subtasks that each illustrated important lessons. Students worked in two person teams so that they would learn to work collaboratively, including coordinating schedules to provide coverage and using documentation and related tools for intra-team communication.

The semester's tasks were divided into four phases:

Introduction

At the beginning of the semester students were introduced to tools for tracking and documenting their work. They were shown how to use a *ticketing system* to enter tasks to be performed, identify and accept tasks that needed to be done, enter information about the task while working on it, and mark tasks as having been completed.

Next, the students looked at methods for creating and maintaining documentation. They learned that wikis are a widely used tool for collaborative systems documentation (Limoncelli, Hogan, & Chalup, 2007) and were encouraged to set up a wiki for their documents. For each task that followed during the semester, they were expected to add to or update appropriate documentation. In particular, they were required to maintain logs that tracked the administrative state of each server or homogenous group of servers and to maintain an operations document (Limoncelli & Grace) for each discrete service.

Basic Systems Administration

Students began working with actual systems by setting up two running servers. They set up Microsoft Active Directory Directory Services for user management and then configured a Debian GNU/Linux server to run a MySQL Database Management System (DBMS). The Active Directory server would be used later in the semester to authenticate users of the web application, and the MySQL server would provide the database services for it. At this point the learning goals were to become familiar with basic tools, to work methodically, and to document work as it was performed so that their documents reflected as nearly as possible the running state of their systems. At the end of this phase students had their first formal assessment, described below.

Advanced Systems Administration

Later in the semester students deployed their web application servers which, together with the Active Directory and database servers prepared earlier allowed them to provide the full application. Students also set up servers to provide backup and recovery services, system monitoring, and configuration management. Each of these tools is described in the *Tools* section below.

Ongoing Operations

Once the full infrastructure was in place, students had to operate and maintain their servers. They were presented with a variety of tasks and problem scenarios, some of which are described in the next section.

Dynamic Tasks

Since practical systems administration involves responding to unexpected events, often occurring at inopportune times, these situations were modelled in the class by introducing *dynamic tasks*. The lecturer had a collection of tasks prepared to be introduced into the class at various times as deemed appropriate to maintain the simulated workplace theme. It was important that students not know the specific nature of the tasks in advance, although they were told to be prepared and appropriate working practices were identified and highlighted to allow them to do so. Examples of dynamic tasks include:

Sick Day: On a chosen day, but without prior notice, a student team may be directed to hand over control of their systems to another team. The other team is then directed to execute a task using only the documentation and resources provided by the original team.

Load Spikes: One reason for using a web application is that it is easy to generate load on the system using tools like Apache Bench. These tools can be used to generate an unexpected load on the system, forcing students to deploy additional servers on short notice. This tests their use of automated monitoring and configuration management.

Security Breach: The instructor places rootkits (Bradley) on students' servers, sets up rogue FTP servers, and uses them to distribute files. All of this is done without informing the students. They then have to detect the intrusion and take action to mitigate the damage.

To facilitate preparation for these dynamic tasks, the lecturer prepared a collection of scripts that performed required steps on each teams' servers. This sort of automation allowed the lecturer to stage realistic working scenarios efficiently and may be the subject of a future paper.

Tools

An important premise of the course was that students should work with the same tools that they would expect to use in industry. Since these tools are generally either open source or are licensed under favourable terms for teaching use, this proved easy to do.

One critical tool for successful delivery of the course was virtualisation (Hickson, 2008) . The only practical way to give a large number of students full control of several servers was by virtualising them. Since virtualisation is itself a real tool used in industry, this requirement fit naturally into the scheme for the course. Otago Polytechnic had VMWare's vCloud Director available, so this was used. A cloud service provider such as Amazon Web Services would also be suitable.

A ticketing system allows users to enter request for tasks to be completed, tracks the tasks as work on the progresses, and manages communication between involved parties. Ticketing systems are an important time management tool for systems administrators (Lear, 2011). Request Tracker (RT) (Best Practical) was used for ticketing. Most interaction with the system is handled through email, so it can be inserted naturally into a typical task management workflow.

Students were required to use a wiki or other shared online document system for documentation and a source code control system for managing scripts and configuration files, but were left free to choose which to use. The intent was to give the students an opportunity to evaluate options and select ones that suited their requirements, but in practice this led to difficulty in assessing student work. During future instances of the class students will be directed to use specific wiki and code management tools.

A backup management system performs scheduled backups, manages storage media, and executes requested data restore operations. Bacula (Sibbald) was used for backup management in the class.

Puppet (Puppet Labs) was used for configuration management. Students defined various server roles and identified the software and configuration required for servers to carry out those roles. The Puppet system then installs the specified software and performs the required configuration on servers. It was chosen primarily because of its cross-platform compatibility, but also because of its wide use in industry.

Nagios (Nagios Enterprises) is a well-known system monitoring package that analyses system uptime and resource usage and that can alert systems administrators to problems and potential issues, and that produces reports on system performance. Nagios was used so that students could be presented with various problem scenarios, identify the problems, and respond to them.

For the web application the primary requirements were that it should be reasonably complex, work with an external DBMS, be able to authenticate users against Active Directory (but not itself be a Microsoft product), and be well documented. Atlassian Confluence (Atlassian) met these requirements and was used, although many others would have worked as well. A MySQL database server and an Active Directory authentication server were used to support Confluence.

Assessment

Limoncelli and Grace list 32 yes/no questions that measure an operations team's compliance with industry best practices (Limoncelli & Grace). Students were introduced to the list in the first lecture and the questions were discussed. These questions set the standards against which student work was assessed. While not every question was applicable to the systems used in the course, most were. For the ones that were not, identifying them and explaining why they did not apply was still informative. Examples of relevant questions include:

- Is your team's code kept in a source code control system?
- Does each service have appropriate monitoring?
- Are your backups automated?

Throughout the class, topics discussed and tasks performed were linked to the appropriate questions on Limoncelli and Grace's list.

Students performed three assessed activities during the semester.

Individual Server Configuration: During the first part of the semester students configured two servers, one running Windows and one running Linux, and prepared supporting documentation and resources. They then submitted their servers and documents for assessment. At this point the complexity of the systems administration tasks that students had performed was quite low. What was assessed was the use of sound workflow: that tasks were tracked in the ticketing system, carried out on the servers, and accurately documented. While students were able to configure the servers, several of them appeared to have difficulty incorporating the ticketing system into their workflows and some tickets were not properly closed. In future instances of the class it may be necessary to provide more guidance on the topic.

Managed Server Configuration: After the first assessment students moved on to build a complete managed and monitored systems infrastructure. They were then assessed over a two week period during which they had to operate their systems while also responding to a number of challenges like the dynamic tasks described above. Students were evaluated on systems uptime, observed performance, team task management, and maintenance of correct documentation.

All student teams, with the exception of one team that did not fully participate in the activity, maintained service uptime aside from periods when the lecturer deliberately interfered with services as part of the assessment. Two teams achieved 100% uptime because their server monitoring was thorough enough to detect and halt this interference before services were affected.

"Final Exam": A traditional final examination would have been inappropriate to this class since an exam would not be part of a typical workplace. Instead, the simulated work experience theme was taken to its logical conclusion. The students were "made redundant" at the end of the semester and had to apply for new jobs. To do this they prepared CVs and cover letters describing their experiences. Then as part of the application process they were directed to take a short written test followed by an interview (with the lecturer) in which they were asked to describe in detail the work they performed, the methods, and the tools that they used. Besides assessing the students' understanding, this task also provided an opportunity to learn useful job searching skills.

As part of their interviews students were asked about systems administration software they had used. Several students commented positively about their experience using Nagios and observed that it proved very useful in operating their systems.

Conclusions and Future Work

To gain some insight into students' perceptions of the course, 24 of the 34 students enrolled in the class were surveyed to determine their views. They were asked to rate the technical level of the class on a 5 point Likert scale.. The mean response for this question was 2.96 ($\sigma = 0.86$). Students were also asked their opinions about the technical level of the paper using a similar five point scale. The mean response was 2.88 ($\sigma = 0.68$). Thus, students generally found the level and pace of the class to be appropriate.

One challenging area encountered in delivering the class was designing, communicating, and evaluating appropriate assessments that fit with the simulated workplace model. For example, a student's success in operating a service may be evaluated by measuring observed uptime and comparing and it with targeted values. However, it is difficult to determine a reasonable value for uptime targets until more students have been observed performing the activity. Another value that can be measured is time to resolve system problems (that are introduced as dynamic tasks), but again it is unclear how long students at this level should take to resolve the problems. Some ideas of target values for these assessments may be found by looking at service level agreements (SLAs) used in industry. These agreements indicate that systems should provide uptime percentages greater than 99% (Baset, 2012). However, it is not obvious how to apply these industry measures to student work. While the assessments used in the first instance of the class worked reasonably well, there is room for additional development of assessment activities and these may be the subjects of future work.

Another problem was presenting an effective simulated workplace model during the first weeks of the class. Students built up their server infrastructures from scratch. Because of this, the tasks performed by students in the first part of the semester may have been too easy, and the volume of tasks may have been too low. This is also not representative of a typical workplace in which systems administrators have to maintain existing infrastructure while simultaneously designing and deploying new systems. In future instances of the class students will begin by taking responsibility for a preexisting virtual infrastructure to which they will add systems and services over the course of the semester.

In general the class seemed to succeed in providing relevant and realistic learning experiences for its students. One student who was already working in a systems administration role commented, ``It's teaching me practical skills that I am already implementing in my workplace."

References

Atlassian. (n.d.). *Team Collaboration Software | Atlassian Confluence - Trusted by Thousands*. Retrieved April 21, 2013, from http://www.atlassian.com/software/confluence/overview/team-collaboration-software

Baset, S. (2012, July). Cloud SLAs: Present and Future. *ACM SIGOPS Operating Systems Review*, 57-66.

Best Practical. (n.d.). *RT: Request Tracker - Best Practical*. Retrieved April 21, 2013, from http://bestpractical.com/rt/

Billett, S. (2011). *Curriculum and pedagogic bases for effectively integrating practice based experiences.* Australian Learning and Teaching Council.

Bradley, T. (n.d.). *What is a Rootkit*. Retrieved April 21, 2013, from http://netsecurity.about.com/od/frequentlyaskedquestions/f/faq_rootkit.htm

Cisco Systems. (n.d.). *Certifications - IT Certification and Career Path*. Retrieved April 21, 2013, from http://www.cisco.com/web/learning/certifications/index.htm

Hickson, G. (2008). An overview of vitual machine (VM) technology and its implementation in I.T. student labs at Utah Valley State College. *Journal of Computing Sciences in Colleges*, 203-212.

Lear, C. (2011, February). System Administration Soft Skills. *Communications of the ACM*, 52-58.

Limoncelli, T., & Grace, P. (n.d.). *The Operations Report Card*. Retrieved April 21, 2013, from http://www.opsreportcard.com

Limoncelli, T., Hogan, C., & Chalup, S. (2007). *The Practice of Systems and Network Administration.* Addison-Wesley.

LOPSA. (n.d.). *Why Join LOPSA?* Retrieved April 21, 2013, from http://lopsa.org/why_join

Microsoft. (n.d.). *IT Certifications | Microsoft Learning*. Retrieved April 2013, 21, from http://www.microsoft.com/learning/en/us/certification-overview.aspx

Nagios Enterprises. (n.d.). *Nagios - The Industry Standard in IT Infrastructure Monitoring*. Retrieved July 1, 2013, from http://www.nagios.org/

Nemeth, E., Snyder, G., Seebas, S., & Hein, T. R. (2001). *Unix System Administration Handbook.* Prentice Hall.

Patrick, C., Peach, D., Pocknee, C., Webb, F., Fletcher, M., & Pretto, G. (2009). *The WIL[Work Integrated Learning] Report: A netional scoping study.* Austalian Learning and Teaching Council.

Pilgrim, C., & Koppi, T. (2012). Work Integrated Learning Rationale and Practices in Australian Information and Commumnications Technology Degrees. *Proceedings of the Fourteenth Australasian Computing Education Conference*, 25-31.

Powell, N. (2012, October 4). *Re: [cplug] seeking software recomendations*. Retrieved April 21, 2013, from http://article.gmane.org/gmane.org.user-groups.linux.cplug.general/14882

Puppet Labs. (n.d.). *What is Puppet? | Puppet Labs*. Retrieved April 21, 2013, from https://puppetlabs.com/puppet/what-is-puppet/

Red Hat. (n.d.). *Red Hat | Training Certifications*. Retrieved April 21, 2013, from {http://www.redhat.com/training/certifications/

Sibbald, K. (n.d.). *Bacula, the Open Source Network Backup Tool for Linux, Unix, and Windows*. Retrieved April 21, 2013, from http://www.bacula.org/en/

VMWare. (n.d.). *VMWare Virtualization - Optimize IT Resources with Virtual Technology*. Retrieved April 21, 2013, from http://www.vmware.com/virtualization/what-is-virtualization.html

Acknowledgements

Thank you to Patricia Haden of Otago Polytechnic for her encouragement and assistance in preparing this paper.

Author contact details:
Tom Clark, tom.clark@op.ac.nz

Please cite as: Clark, T. (2013). The Introduction of an Advanced Class in Systems Administration at Otago Polytechnic. In H. Carter, M. Gosper and J. Hedberg (Eds.), *Electric Dreams. Proceedings ascilite 2013 Sydney*. (pp.144-150). https://doi.org/10.14742/apubs.2013.1376