The Uptime challenge: A learning environment for value-driven operations through gamification

KYRRE BEGNUM, Oslo University College of Applied Sciences
STIAN STRØM ANDERSEN, University of Oslo

This project aims to incorporate a specific perspective from the profession of system administration into a course taught at BSc level. The perspective is that of value-driven operation. Students are given access to a voluntary game which resembles a realistic scenario where they will manage a website given to them in a real cloud environment. The students earn points for the amount of time the website is kept up despite several automatic obstacles introduced by the game system. The goal is to let students reflect about the strategies that they use in order to optimize site resilience and how that leads to increased value generation.

1. INTRODUCTION

System administration is a profession that is still in its infancy world-wide. With society’s dependence on IT systems, the role of the system administrator changes from “IT guy” to “guardian of vital civic functions”. Becoming a system administrator does not require formal training or certification, yet it can be argued that in some cases their role is as important for society as nurses, teachers and law enforcement, all of which are professions with certification demands and corresponding education programs.

There are only a few undergraduate programs that focus on system administration. They are commonly formalized as computer science programs with topics belonging to the technical side of system administration, such as: operating systems, security, network protocols, scripting and configuration management. These courses fit naturally within the computer science umbrella, but given that system administration as a profession has a heterogeneous knowledge-base, there are topics that are missing. Examples of such topics are: Organizational structures and leadership, communication skills, technical writing, economics, law and ethics. System administrators apply general computer science on a specific case which is the combination of IT systems and the organization depending on them. An education towards that profession should therefore encompass skills and knowledge that will make them successful in that setting.

Established professions, such as nursing and teaching, are characterized by a value system and identity that is adopted and internalized by the candidate as part of their education. It can be argued whether, at the time of writing, this identity is in place or even properly defined for the profession of system administration. Evidence can be found of activities that aim to identify this identity, but as a whole this process will require more time. One example was a workshop named "State of the profession" which was organized in conjunction with the USENIX Large Installation System Administration conference in 2012[USENIX 2012]. Here, the topic of lack of identity was lamented by the community, especially as it hurts cooperation with other professions and users in general, such as economists, developers and management. A lack of identity makes it hard to manage the expectations from the rest of an organization. In organizations such as LOPSA (League of professional system administrators) the topic on what it means to be a system administrator is still actively discussed in member forums. LOPSA also released a code-of-ethics that is a good example of an effort to represent the values that are idealized by professionals[LOPSA 2003].

In Limoncelli’s The practice of network and system administration some chapters are devoted to what it means to be a professional system administrator, such as Perception and visibility and Being happy [Limoncelli et al. 2007]. In 2014, Kim et. al published the novel The phoenix project: A novel about IT, DevOps, and helping your business win used a narrative style to show a software project gone awry from the perspective of a system administrator thereby providing the reader with a unique insight into his value system and professional codex. Given the novels popularity and reception, there is reason to believe that it captures some of the spirit of the profession[Kim et al. 2014].
From an educator’s perspective, understanding the profession one is representing is difficult when the profession itself is so new and not well defined. However, through the process of defining learning outcomes, other educators have created models that make it easier to define and map the professional beyond their basic skills. One such inspiration can be found in Harden’s work on characterizing learning outcomes for medical education. Harden creates a model of circles where the innermost circle describes doing the right thing followed by an outer circle of doing the thing right which speaks to the quality and standards of the application of the correct method. Finally, the outer circle is named the right person doing it which addresses the standards and value system of the individual [Harden 1999b; 1999a]. The same model can well describe the profession of system administration from a profession perspective. However, when we look at current education programs, only the innermost circle is the focal point.

One reason for the lack of holistic focus may come from the fact that there is little literature yet on what represents the outer circles of Harden’s model that are written for the system administration professional. There are a few examples of chapters in books that deal with some of these issues, like mentioned above, but this is insufficient as a base for a course in any of these topics. From a teacher’s perspective, this is a challenge. We understand that there are certain unspoken truths in the profession that are not part of the student literature. This value-system, that is transferred in an unstructured way, either through co-ops or when the candidate starts working somewhere, is what educators want to teach in the same way the technical issues are taught. Harden’s model is intuitive and simple enough to start developing descriptions of skills, behavior and attitudes that make up the profession so that values and their focus can be made more clear in the development of learning outcomes in system administration programs.

In this project, we focus on one such professional perspective from the profession that is rather new and usually overlooked. We will attempt to create a teaching environment for it so that students may learn it as part of their education. The concept of interest is that of Value-driven operations which is our own term for the concept. The idea is that the performance of an operations team is seen as a result of the amount of value it helps the organization to produce. One might be quick to assume that money is the obvious value, and this is certainly a strong driver for many organizations, but there are other examples. Consider volunteers working for a non-profit organization that has just had a major breakthrough in media due to a natural disaster or humanitarian effort. The operations team will increase the organization’s value, if they manage to keep the web-based donation system running throughout the surge of donations that come in. Systems that deal with public outreach may have a higher value if they manage to remain accessible in times of crisis. An operations team also provides value, if it enables the developers to produce and put into production software features to their systems that either increase profit or drives competitiveness. In essence, any time an organization sets forth a mission, IT must be able to relate what they do and why to how it helps the organization achieve it.

Keeping a value-driven-operations mindset means the following:
— Understanding their “value” in what the organization strives to achieve
— Identifying the processes in the organization where IT is involved and their impact on value
— Enabling continuous improvement to the processes so that value can be achieved more often, effectively or predictably

Knowledge about value-driven operations provides an alternative perspective to how IT works in an organization to the more traditional views of striving for security or achieving the highest possible performance. There is no “best” in these perspectives. In our view, the right person doing it, to use Harden’s phrase, is one who can balance all of the perspectives in the light of a particular context. Security and performance are topics often discussed while value-driven operations is virtually absent.

2. ORGANIZATION OF EDUCATION
We wanted to create a teaching environment that had the following attributes:
— Increases student motivation
— Enables self-assessment and self-regulation
— Creates engagement and foster discussions in class
— Provides a direct feedback loop between the students actions and value creation
— Has industry relevance

In order to address the first item, we want to use the concept of gamification. Gamification means incorporating learning activities through games and can increase motivation by providing more fun and engaging ways to work with the subject.

The game in this case will be in the form of a simulation, were the students are divided into groups of two. Each group represents their own company that is in charge of their own web-page. The company also makes money as long as the web-site is up and can be used. In other words: from an operations perspective, the value of the organization is tightly coupled with the uptime and resilience of the web-site.

The groups will get the web-site as downloadable software and have to make the site run in their environment. They are not allowed to change the code of the site itself, a typical scenario for the business world. When they go “live” with their site, traffic is sent to the site creating a sense that something is happening and is the beginning of their simulation.

However, things happen to their site that may push the site into a faulty mode, at which point the company loses money. Bugs in the code, misconfiguration and outright automated sabotage will challenge the students to come up with architecture designs and develop their own tools that counteract these. What they spend their time on improving has to be constantly balanced based on how much more money they will make with the improvement and what they should do first.

Throughout the simulation, the amount of money earned is visible to all students at all time. This means that while they work, they can see the effect of their (and others) efforts. For example: a group decides to have multiple webservers in order to increase the redundancy of their infrastructure. After the task is finished, they suddenly realize that one of the two web servers has been shut down by the automated sabotage system. However, they note that they didn’t take any damage financially as the other webserver kept serving webpages throughout. They discuss what happened and decide to also create a tool that can automatically starts servers that are shut down so that the site can grow back to it’s intended state without human interaction.

The simulation is close to real life as development and operation are traditionally two separate groups with distinct goals. Designing a scaffolding of robustness around a shaky software solution is unfortunately a common situation and one that candidates should be prepared to tackle. Furthermore, the advent of cloud platforms such as Amazon AWS, has led to the deployment of massive services such as Dropbox and Netflix that run on clouds that do not guarantee for the safety of their virtual machines. This increases the need to create an infrastructure that can generate value through resilience.

3. THEORETICAL BACKGROUND

For profession-based educations, the balance between theoretical subjects and practice is an important and reoccurring subject. Programs curricula are often torn between vocational skills and abstract theories. Good arguments exist for either perspective, but the challenge is the continuous balancing of the two in light of changes in the field. Decreasing the gap between theory and practice for students is known as creating coherence in the program[Hatlevik and Smeby 2015].

A high level of coherence means the students experience a strong connection between theory and practice. For system administration education, this balance is often found in the combination of theoretical lectures, for example networking protocols and operating system concepts, and lab exercises such configuring a network and installing operating systems respectively. For the case described in this paper, we aim to create a similar experience. Discussing value-driven operations in class is highly theoretical and unless there would be an environment for experiencing it first-hand, the students might have a low coherence as the topic is not joined by a practical application.
Hattie et al. argues for the use of feedback mechanisms as a method to increase the output of a teaching methodology [Hattie and Timperley 2007]. In their model, they argue that feedback can be given on four levels: Task level, Process level, Self-regulation level and, finally, Self level. The effectiveness of the feedback is relative to the level it is given on. For example, feedback on the self level, like “you are a good person” does not increase performance as much as feedback on the lower levels. In the framework we propose, students will get immediate feedback on their tasks through the automated scoring system. If they do a good job implementing a technical task, the score will automatically continue to increase while the opposite will happen if they make a mistake. According to Hattie, this is a common used strategy, however in our case it is automated so that it does not depend on a teacher or assistant to provide the feedback. Feedback of this kind is the most effective when it is coupled with information. When lack of information is the cause for a mistake, simply pointing out the mistake will not help the student unless the information is also made available. That aspect is not entirely covered in our solution. Students will see some additional technical performance metrics that can help them interpret their own results, but they need to have sufficient knowledge to use them as feedback.

Process level feedback enables the students to understand how they handled the process of conducting the task. How did they troubleshoot? How did they plan what to do? How did they test that their solution was working? This type of feedback is quite important as it focuses not only if the task was successful, but wether the approach was good or not. This resembles the two inner-most levels of Harden’s model. “Doing the right thing” is related to successfully completing the task. “Doing the thing right” means completing the task in an effective and stable manner. For example: One typical failure will be that a random server will be shut down by the sabotage system. Successfully completing that issue is to simply boot the server up again. That will be the right thing to do. However, as this happens at a regular interval the students will not have solved that problem in the best possible way. They will have to spend the same amount of time repeating that step every time a server is taken down. It is a short fix, but over time it drains time. Dealing with that issue the right way, is to invest time in writing code that automatically boots servers that are shut down, thereby eliminating the need to handle such issues in the future. This type of feedback is present in our solution, but it is not obvious. This type of feedback is only available over time, when the student experience that dealing with reoccurring issues with automation and redundancy will result in less work in the long run.
Using games as a means to motivate students and increase learning has become a popular tool in educations through the increase of more digitized learning environments. One example is the work by Hembroff et. al, where they create a learning environment based on a game in a security class [Hembroff et al. 2015]. The difference in their approach is that the game contains a story and levels and will work on its own. In our case, the game takes place as their environment is exposed to the typical climate of an unstable solution running in an unstable environment. Furthermore, it requires them to do the normal assignments in the class in order to have an environment to work with. So we essentially play with their lab environment.

4. UPTIME CHALLENGE SPECIFICS
The site the students were to manage was a social network type solution written in PHP with MySQL as the database backend. Each student group is a company that will make money based on the uptime and availability of their social network. If the site should be down, the company will suffer financially and loose money. A growing balance is therefore a literal indicator that more value is generated as an effect of their work as sysadmins. Achieving their company’s mission means to maximize reliability and uptime despite a growing user-based and other obstacles. At numerous occasions, they will face problems that challenge them to either do the right thing or the thing right, as Harden would put it.

The code for the site was made available through Github. The site was relatively simple, with users, posts and comments. The site was not meant for actual human interaction, but through scripts that were executed from a simulation engine which will be described in further detail later. The site existed in several versions, each with particular bugs and characteristics. For example, version 1 resulted in several PHP error messages in log files at every page visit. This would lead to a full disk and slow performance once the site grew larger. Another version had no cap on the amount users listed on the frontpage. As a result, as the user-base grew, the landing page got increasingly larger.

The students were instructed to use a particular version well knowing that it had some challenges. At later stages, we simulated upgrades in that students were allowed to use the next version available. The storyline was used that the software was developed by consultants so they had no option to go and modify the code themselves, they had to wait for the consultants to develop a new version first.

The course covers topics such as caching, loadbalancing, database replication, scaling and backup. At every topic, a new component is added to the infrastructure so that what started as a single webserver with a database server ends up as a site with several webservers, redundant load-balancers, replicated and redundant databases, caching servers and backup servers. Throughout this weekly evolution, the simulation engine kept using the site non-stop and increase its users, posts and comments. As any start-up, they had to grow while the site was active. Every site started with a baseline of 50 users, but most groups ended up with more than 30.000 users and over a 100.000 posts.

The technical infrastructure used was an OpenStack cloud. One goal was to be as realistic as possible with the technical setup. OpenStack and Ubuntu virtual servers were a technology base that had a high transferral value to the real world. Each student group had it’s own tenant in the cloud with resource quotas associated with it. Using this platform, we could build our own simulation framework into the same cloud and we also gained access to their virtual servers. The cloud allowed us to track the resource usage of each tenant and this was also included into the simulation. Each company had to “pay” for how many resources they used, so the most payoff was gained from the leanest yet scalable solution, closely mimicking the real world.

4.1. Simulation engine
The simulation engine was responsible for several actions:

— Constant usage of the students sites (landing-page visits, new users, new posts and comments)
— Measuring site availability and response time
— Award/deduct money to each company based on availability and response time
— Sabotage the companies sites by randomly shutting down students servers
— Deduct money based on the cost of a company’s infrastructure

The students sites usage is based on 24-hour profiles with traffic variation both based on time of day and with an additional noise factor to allow for small spikes. The amount of usage is calculated by the simulation engine for 5-minute intervals and sent as a job to a group of workers. An example of one such job would be that Company X got a sustained rate of 7 landing page visits per second for the next five minutes. Traffic would peak around work-hours and be low otherwise. In order to reduce the pressure of the job workers and cloud overall, each company existed in a different time zone so they would peak at different times during the day.

Each company’s site was checked for availability every five minutes. If the site loaded successfully and a particular text was found on the page (a text that would only be there if the site functioned properly i.e. the database etc. worked.), the company was awarded a base amount of money. An additional bonus was awarded in case the site loaded in less than 0.5 seconds. The bonus declined rapidly in case of a slower loading time. If the site was available, but did not work probably (typically the text was not found because the database was down, or the students had redirected traffic to an emergency page), then a small fraction of the base amount was awarded. If the site would not answer at all, then the base amount would be deducted from their balance instead.

Inspired by Netflix’s Chaos Monkey[Tseitlin 2013], we also introduced sabotage to simulate an unpredictable infrastructure. The sabotage was relatively benign in that it did not destroy anything the students had done, but merely shut down virtual machines. Every two hours one random server belonging to each company was shut down. The effect could be dramatic if the company lost their single webservice in the middle of the night. Doing the right thing, would be to start the server again in the morning with the losses that would entail, but a group that had either invested in writing a small script that checked for downed servers would maybe only get a very small penalty. If they had redundant webservers, they may not get a penalty at all. These are examples of putting Harden’s three circles into practice. The two latter strategies would mean doing the thing right and with a small investment of time save time later and increase value. For convenience, the sabotage was disabled during lab hours to not interfere too much while students were working actively on the course. On some occasions, the sabotage was more manual. For example, at one point we logged into all loadbalancers, shut down the loadbalancing service and removed the configuration file so the service would not start again directly. These types of occurrences were intended to facilitate troubleshooting.

Money deduction based on infrastructure size was based on the same principles as Amazon AWS: you pay for uptime factored by the size of the virtual server. The main difference was that the cost was calculated every 5 minutes and only for each server that was in fact up in that period as opposed to an hourly charge regardless of system uptime in AWS. This was to encourage scaling and a more dynamic system.

4.2. Simulation infrastructure

Development of the simulation architecture was completed as part of a MSc project. Our goal was to create an extensible framework were several features could be added at a later stage and could potentially serve other use cases in different courses at the same time. For example, we envisioned that in the course there would be periods of increased traffic, due either to a trending topic or campaign. During these periods, the traffic would be scaled up and the potential risks and rewards likewise. So we wanted to create something where these types of additional components could be added at a later stage.

The architecture was based on the idea of individual service managers for each sub-component utilizing a common framework of database access and job queues. The framework itself had an API available to all sub-service managers in order to interact with the data layer. A manager would be in charge of one set of actions, for example, deciding the amount of traffic for all sites at any point in time. The manager would create jobs that are sent out on a message queue where corresponding
workers waited for jobs. The managers would also interact with the database layer. In some cases for information retrieval, like fetching the corresponding IP address for a company’s site, or information changes, like awarding more money. Each individual worker job ended up with a report that was stored in the database. This enabled students to look at the history of actions as well as us to find evidence of student activity.

During the run of the course, with 15 student groups/companies, we used 15 virtual servers hosting job workers and three other servers with database, API and managers respectively. Using message queues, the architecture could scale based on necessity and our desired traffic workload.

Students could get feedback on their progress through plots and score listings that were made available to them and were continuously updated. During the course, some students wrote alternative dashboards that had better visualizations and graphics. One student even wrote a mobile app to track the state of the site and its current company balance based on our data. Students also set up their own monitoring and in one case there was a friendly dispute where our system had flagged a site as down while their data said it was up. All of these cases were strongly welcomed by the staff and introduced not only interesting problems to discuss in class but also was interpreted as signs of student engagement.

5. CASE ASSESSMENT

Due to the nature of the course, there is no option to conduct a test/control study in order to establish the success of using this uptime challenge. Even though conducting a comparative experiment can be valuable in a statistical setting, its practice in an educational setting has been criticized by Freeman et. al [Freeman et al. 2014] in cases where there is strong reason to assume that it will be of benefit. They point out that if there is a underlying understanding that one teaching method is more advantageous to the student, offering it only to half of them in a test/control setting is unethical towards the students in the test group, where such methods are not used. Furthermore, as the game will go on for most of the duration of the course, it would mean that in a comparative study, half the class would witness as the other half would be having special discussions and participate in a game that is meant to be fun. Thus, the decision was made to let all students partake and rather measure learning on all students throughout the course.

The data collection was done in the form of questionnaires conducted in the beginning and end of the semester as well as reviewing work they hand in during the course. The questions were originally asked in Norwegian, but have been translated to English here for the sake of this text.

Throughout the course, the data traces from the management system itself will be available for analysis. This enables us to track student engagement by observing the degree of points scored throughout the semester.

6. RESULTS

The uptime challenge was utilized in the course “Database and application management” at the Norwegian Technical University during the spring semester of 2016. The course is a mandatory course for the BSc in Network and System administration as well as an elective course for other computer science students.

As the class started, the initial survey was answered by 25 students of which nine of the students had some prior industry experience. 8 of the students had this course as a mandatory course, the rest took it as an elective. 19 of the students (76%) agreed or somewhat agreed to the statement “I am motivated by working on course activities outside the mandatory tasks.” When asked about what activities they believed were most central to their own learning, being at lectures, trying out things for themselves and working together with others were the strongest ones reported.

The survey also asked the students to explain how they understand the term “System administrator”. Almost all students wrote a definition that relates either to being “the person responsible for all the systems” or listed specific tasks, like “In charge of securing the infrastructure” or “Monitors that everything is up”. None of the answers were connected to the contribution that the role has to the a company’s mission. However, when asked about what additional properties besides being...
“secure” would make a solution be considered as "good", the students focused more on the softwares role, like "employees can work more effectively" or "User friendliness". Furthermore, when asked in what ways they believe an IT solution can add value to an organization, the vast majority of the answers centered around the increased effectiveness of the users and the resulting potential for money savings.

Overall, the data suggested that most students had a technical and low-level attitude towards IT systems and being a system administrator. They did not see a system administrator as someone directly connected to a company’s mission, but rather someone who keeps the lights on. This was expected as this is a prevailing attitude towards the profession.

After the course, we conducted a survey where we got 23 replies. We asked the students to what degree they agreed to the following statement: The uptime challenge motivated me to spend more time on the course. The alternatives were on the following scale:

— Do not agree
— Somewhat disagree
— Neither agree nor disagree
— Somewhat agree
— Agree

All but one student either agreed or somewhat agreed to the statement (11 on each). Furthermore, the students were also asked whether they agreed to the statement: The uptime challenge game motivated me to use more time on activities related to the course that were not directly to the uptime challenge itself. Here, 16 students either agreed or somewhat agreed, while five neither agreed nor disagreed and two somewhat disagreed.

The students were also asked the same questions from the beginning of the course, as to how they understood the term “System administrator”. There was no particular change from the beginning of the course, but further inspection showed an increased perception of what the role meant, rather than what the role does. Of the 23 respondents, 14 used the term “responsible” or “responsibility” when describing a system administrator. Only seven students used that term in the first survey. There were less examples of tasks in the end-survey than in the first and more on achieving service availability and ensuring effectiveness of workers. This may indicate that the opinion of the system administrator changed towards the effect a good system administrator achieves in the organization rather than the particular tasks they do.

### 6.1. Tracking individual groups performance

Overall, the data traces from the worker reports showed that the majority of the groups earned money most of the time. In order for this to happen they have had to implement automated mechanisms to recover from the sabotage, which were not mandatory tasks given to them. In addition, they had to compensate for the growing size of the site by adding more capacity to their infrastructure. However, we found examples of groups that did not attempt to optimize value. They would normally boot up all the downed servers in the morning of the lab-day and then start loosing them again the day after when the sabotage started again. However, no student voiced direct negativity towards participating in the challenge. Also, it was not investigated if students wanted to participate but did not have the time to do so. We also do not know if they attempted to improve the robustness but were simply unsuccessful in doing so.

As an example, we present the distribution of two groups that had different performance profiles. For simplicity, we call them group A and B. Group A did not implement any automation and as a result ended up with low rewards and often deductions. From the distribution plot we can see how the distribution is almost uniform. The reason the rewards are often higher is due to the added bonus for response time. Group B took great interest in the challenge and often experimented with

---

1The questions in all surveys were in Norwegian, but have been translated to fit the language of this paper.
alternative technologies to squeeze the most performance out of their site. As a result, one can see from the distribution that they achieved almost entirely high rewards.

7. DISCUSSION

From the results of the survey we can argue that the uptime challenge was something that motivated students to show effort in our course. This aligns with our impression from teaching also. The fact that several students developed their own tools and apps to better get a notion of the current financial ranking and status of their company, tells us that the challenge was engaging. The uptime challenge created a storyline and scaffolding on which most topics in the course could be pinned.

7.1. Achieving a higher-level view of a system administrator

Our initial main goal was to connect some more abstract ideas about value creation into a framework that let the students experience how it relates to actual technical tasks and strategies. Throughout the period, and based on the predominance of the students low-level view of what the term system administration we started to ask ourselves if our uptime challenge will also influence their perception towards a more high-level view, like that the system administrator is essential in making sure the IT infrastructure helps achieve the organizations mission. Given the increase in the word “responsibilty” one might be hopeful of that, but it was not explicitly researched.
We see, however, how we could tailor the challenge to include more organizational thinking and interaction. For example, the students could have opportunities to argue for increased budgeting and resources. We could also have specific goals that had to be achieved at a specific point in time, like a new release or campaign by their company, where uptime would be of particular important to the companies mission.

7.2. Availability of software
All the software written that is part of the uptime challenge is available as open source. This includes the web-site used by the students and the simulation infrastructure. The tools are continuously improved to help teachers easily manage student groups in the system.

However, there is certainly a cost associated with the uptime challenge. As will most lab environments, maintaining the lab often times falls on the teacher and a lab engineer. Our experience was that setting the system up in the beginning of the class was the most time consuming period. Once the system was up and groups were added to it, it was mostly a matter of scaling for optimized utilization. Through the use of Docker containers and the puppet configuration language, scaling was not a time consuming task.

Being dependent on a cloud or other virtual infrastructure is a potential limiting factor. We have not investigated how one could split the solution so students could run the challenge on their own computers. We definitely see the potential value for this in terms of self-learning and this could be further investigated.

8. CONCLUSION AND FUTURE WORK
This project wanted to create a lab environment and simulation that enables students to reflect over the relationship between their operations work and the value that is generated by the managed solution as a result of it. Our implemented lab environment is a adaptable solution that can work for several different types of internet-based sites as cases and has a high degree of customization. The environment was tested in a course and surveys at the beginning and end of the course were used to track students satisfaction and attitudes. Our results show that students considered the environment to be engaging and motivating. Most students took advantage of the opportunity despite it being an optional activity in the course. Future work will focus on making the solution adaptable and usable by other institutions and investigate if a self-study solution can be developed from it.

REFERENCES


Gene Kim, Kevin Behr, and George Spafford. 2014. The phoenix project: A novel about IT, DevOps, and helping your business win. IT Revolution.

