platform.sh

WHITE PAPER | SAVINGS OVERVIEW

February 2015

INTRODUCTION

This document provides anecdotal, testimonial and statistical evidence directly related to the implementation of Platform.sh, which offers unique new capabilities that significantly optimise the code development process and improve production hosting services.

The savings and gains described come from 15 different organisations running online businesses, and 8 Digital Agencies (DA) providing web-site build and deployment services to multiple sectors. The DA's are largely Drupal/PHP specialists with delivery teams numbering between 8 and 90. **We provide 14 different data points ranging from €14,000 / $16,000 per annum to €350,000 / $400,000 per annum, although most customers benefit from savings in multiple areas, some totalling many hundreds of thousands each year in business benefit.**

These savings are achievable through the Platform.sh approach to the management of infrastructure, which has a huge impact on developers, allowing them to branch new environments on-demand and at no extra cost. When given the choice, developers are using this facility for very high frequency branching, because it ensures they are always working in the latest copy of the target environment, which guarantees zero problems or surprises when they merge tested code upstream. With Platform.sh, submitting tickets for new environments or configuration changes, along with long waiting times for DevOps are a thing of the past. We are seeing developers moving from one new environment per month to hundreds, and consequential improvements in the delivery of high-quality code.

There are many other areas of the development process that benefit here too, including on-boarding new developers, contractors and third-party vendors; shifting the profile of skills required on a project, and automating the Continuous Integration (CI) process from end-to-end.

Production hosting service savings are based on our seamless scaling capabilities, which have saved many sites from becoming unavailable during unexpected peak traffic periods.

And finally, the combination of fast branching and highly redundant production service levels offered means new levels of Continuous Delivery (CD) can be reached. Very fast changes can now be made into production with a high degree of certainty that they will work as expected, which means you can now optimise your site when you're at your busiest, and see the results instantly: which is nirvana for the sales and marketing teams!

OUR DELIVERY EXPERIENCE ACROSS HUNDREDS OF PROJECTS

Platform.sh was conceived by the technical leadership at Commerce Guys in 2013, who, after delivering hundreds of Drupal eCommerce projects, were seeing many recurring problems in the development processes which were also common to many of their partners around the world. Platform.sh is a second-generation Platform as a Service (PaaS) that automates the majority of these constraints out of the process, thereby liberating developers and technical management.

Not only have we built and delivered over 250 Drupal applications around the world since 2011, we provided support for them, too. Our web site provides further detail on our 24/7/365 application and site support services if you're interested.

HOW VALUABLE IS PLATFORM.SH TO A DIGITAL AGENCY

The business management objectives of a Digital Agency, Systems Integrator or in fact any technology business delivering and maintaining applications are threefold: improving the quality of what you do; reducing the time it takes to do it; and at the same time raising your project gross margins.

Whilst large organisations may take longer to justify changing their approach to delivering applications, or adjusting the delivery process regime, agencies and integrators need to be more flexible and adaptive. After all, your ability to do this better than the customer and the competition is what keeps the business winning new deals and growing bigger.

When your entire business is based on the quality and speed with which you deliver your customers' working applications, making some small changes can have a big effect.

Platform.sh is helping agencies do all the following:

- Lower developer costs in the initial build and maintenance phases
- Accelerate the application delivery cycle
- Reduce the footprint and lower the cost of maintaining the server/storage infrastructure
- Introduce end-to-end CI and offer true Continuous Delivery that will win you contracts
- Make double-digit gross margin improvements at the project level

In addition, we have many customers making public reference to how valuable Platform.sh has been for their business, and that they wouldn't have been able to make the changes they have done without it, or that that Platform.sh saved their sale and so on.

As for implementing Platform.sh into your environment, we have made it easy to use and it's constantly getting better. It easily integrates with your current tool-sets, and we're finding there are very few constraints when migrating existing applications to it. And by way of offsetting any adoption costs, we give you a healthy rebate on all the services you take up, re-sell or introduce us to.

So the final question for you should probably be: '**why haven't these guys given us a demonstration yet?', and 'why aren't we trialling Platform.sh already on a small project?'**

SUMMARY OF SAVINGS ACHIEVED IN 2014

The following summarises the headline savings being achieved by Digital Agencies and Enterprise organisations adopting Platform.sh for PHP-based project development and live site hosting. All these savings are explained in more detail throughout this document.

SAVINGS APPLICABLE TO ALL PROJECTS

On-boarding new developers and providing 3rd party access	"Up to €75,000 savings"
Re-prioritising feature development schedules	Lost billings p.a. by project size: small €18,000, medium €30,000, large €120,000
Use of branching & merging greatly increases	Project savings and gains: small £ 39,750, medium €237,750, large €278,250
Skill-set downshift	Making a 14% cost reduction in developer resources equals 24% gross margin improvement
Continuous Integration (CI) – e2e process automation	"We have 7 Systems Administrators and DevOps, 3 of whom are now doing a lot more billable work" circa. €350
Offshore / remote working model	Estimated 10-15% infrastructure cost reduction and far superior assurance/ governance processes
Infrastructure savings	840 monthly hosting units reduced to 428 with Platform.sh, saving circa. €246,400 p.a.

.

SAVINGS APPLICABLE TO E-COMMERCE

Preventing revenue loss	"We no longer worry about outages during sales campaigns" €100,000
Improving sales	"About €30,000 today" and "Saved us €85,000"

DEVELOPMENT PROCESS VALUE: *ALL THIS DATA IS FROM OUR CUSTOMERS!*

Platform.sh not only optimises the development process, it actually enables the majority of development regimes to change in a much more efficient direction, by:

1. Removing the complexity, costs and waiting time associated with DevOps, and replacing configuration management tools such as Puppet and Chef.
2. Allowing developers to clone full-stack copies of Master environments on-demand, so that they can develop each new feature in an environment 100% consistent with live, thus eliminating the majority of common errors that need time and effort to resolve
3. Implementing advanced Continuous Integration (CI) principles through integration and automation of Test/Behaviour Driven Development processes,
4. Automating deployment processes into the cloud.

SAVINGS DATA AND FEEDBACK PROVIDED:

On-boarding new developers and providing third party access

The costs associated with setting up new developers or giving access to external vendors and consultants is frequently cited as high. Platform.sh clones the full stack environment ready for a new developer to start work within minutes. New developers have read-only privileges on upstream branches so they can get up to speed without any danger of committing code to critical environments.

Typical time taken to set-up new starters	Preventable loss of billings
1-2 man days per person. For a medium sized Digital Agency totalling 50 new set-ups per annum = 100 man days @ €750 per day	Up to €75,000

Re-prioritising feature development schedules (days wasted per project)

The well known global Digital Agency that calculates this as a saving, follows their own unique, non-Agile development process. The nature of their predominantly public sector customers sees frequent changes in budget availability and changing priorities in project features. Of course, every time the plan changes and new feature sets are moved up to the starting line, an intense period of analysis and separation of code occurs prior to switching. Making the right decision to continue on, or regress to an earlier point in the development, saves time and money.

The following table is based on data from 3 Drupal agencies, with team-sizes of 10 people, 40 and 100. We averaged the data they gave us on wasted days by project size.

Project size	Man days wasted p.a	Lost billings
Small	30	€18,000
Medium	50	€30,000
Large	200	€120,000

Use of branching & merging greatly improves the delivery process

The cost and complexity of ‘branching’, creating and managing new environments is usually quite high, and is therefore an occasional practice in most developments (ie. when necessity demands), or non-existent for smaller projects. The irony is however, that the more complex the development, and the faster the pace of change within the surrounding technical environment, the greater the need to develop on clean copies consistent with the upstream target. There is a clear, inverse correlation between the frequency with which environments are synchronised with the target environment, and the number of bugs and time spent fixing them.

The perfect scenario for a developer is one in which they can clone upstream environments at will, so as to continue coding and testing with complete peace of mind that the upstream target for their finished feature bears no inconsistencies whatsoever, and will therefore produce no problems for them once arriving there. The advantages of branching are enormous, and much business benefit is lost when conditions prevent this being done frequently and easily.

Platform.sh of course enables developers to branch and clone the full stack of supported technologies on-demand, at zero cost and in under 30 seconds seconds, no matter how complex the build or how large the environment! The table below demonstrates the huge increase in branching seen amongst development communities taking advantage of this feature to improve their workflow. For teams switching from SVN the effects are even greater, with a migration made easier by the fact that Git is native within Platform.sh and many functions automated.

Larger projects usually operate within a much more complex ecosystem, with a greater number of developers working with a wider set of technologies, deploying into many more environments which all vary by region for example. A further important observation here is the exponential increase in branching within larger and more complex projects! Improving the general levels of consistency within fast changing environments supporting LAMP stack plus multiple other technologies, all being simultaneously worked on by different development teams has an immediately measurable impact on elapsed timescales and delivery cost.

The chart below is based on empirical data from 5 Digital Agencies (DA) with Drupal development teams of between 10 and 100 (including DevOps). The people savings are calculated by removal of expensive SysAdmin activities, and productivity gains introduced by allowing the developers to create environments on-demands at 5-6x prior levels.

Project size	Small	Medium	Large
Current Process			
Time to create new environment	45 Minutes	$ hours	16 Hours
Total spin-ups p.a	100	100	50
With Platform.sh			
Time to create new environment	5 Minutes	.05 Hour	1 Hour
Total spin-ups p.a	500	500	300
Man day effort reduction p.a.			
Savings	32	195	270
Gains	21	122	101
Total man days recovered	**53**	**317**	**371**
Opportunity revenue @ €750 per day	**€39,750**	**€237,750**	**€278,250**

Skillset downshift

Most projects require a range of skills to optimise delivery speed, quality and cost. For example, if you don't have DevOps skills on the project itself, you need dedicated Systems Administrators or tools that automate some of the more complex business processes. Where you are able to, expert and master developers provide planning, design, governance and trouble-shooting across projects, with intermediate and advanced beginners doing most of the coding (see skill level definitions below).

Where there are minimal tools in place - or too few Systems Administrators - much higher levels of involvement are required from your experts and masters. Because Platform.sh automates environmental management, facilitates Continuous Integration (CI), encourages branching and comes bundled with advisory tools for coding and performance, the general level of developer experience required centres around coding skills, and that means a larger proportion of lower level/lower cost skills-sets can be accommodated to deliver the project, as depicted below:

Cost Perannum	Cost per day	Skill level	Allocation BEFORE	Cost	Allocation AFTER	Cost
€75,000	€341	5	10%	€5,250	5%	€2,625
€60,000	€273	4	15%	€6,300	8%	€3,150
€45,000	€205	3	30%	€9,450	25%	€7,875
€30,000	€136	2	40%	€8,400	50%	€10,500
€22,000	€100	1	5%	€770	13%	€1,925
			100%	€30,170	100%	€26,075

The simple project profile in the above table above makes a **14% cost reduction** in developer resources required when using Platform.sh. Within this model that equates to a **24% gross margin improvement** in the professional services line item. For reasons of simplicity, the costs shown are based on salary only, and an average 70% utilisation rate across the development team. The opportunity revenue against billing these resources onto other projects has not been calculated.

Needless to say, the larger the project the greater the financial impact of this down-skilling.

Short skill level definitions, using Drupal as the example

LEVEL 1 **Novice** - At minimum, a knowledge of the PHP programming language, and fundamental web technologies such as HTML. LEVEL 2 **Advanced Beginner** - Proficient in administrative tasks such as installation and configuration of Drupal core and installation of third-party modules. Basic understanding of coding to modify and extend Drupal's behaviours. LEVEL 3 **Intermediate** - Proficient in writing custom Drupal modules. Aware of core contribution, APIs and appropriate use. Knowledge of hooks, core and common modules and how their use affects security, performance and maintenance. LEVEL 4 **Master** - Expertise in most common areas of development. Comprehensive knowledge of APIs. Cost-benefit awareness of various approaches / avoiding common pitfalls / troubleshooting problems and improving codebase quality. LEVEL 5 **Expert** - Authority on Drupal development and delivery planning of large features and entire sites. Intimate familiarity with Drupal and surrounding infrastructure, including LAMP stack and supporting services.

Complexity of the project directly related to the skillsets allocated

A key issue on many Drupal projects is determining the development effort and timing of resources required to deliver them. The scale of resourcing is obviously dependent on the size of the project, but the composition of development resources is dependent on the complexity of the project's features.

The effect of the project category (below) is to push the weighting of skillsets up or down the skillset downshift table (see previous section), impacting project gross margins accordingly.

A. **Basic customisation -** Sites in this category require basic customisation, which can be performed by installing and configuring existing Drupal modules, with visual customisation provided by the use of existing "off the shelf" themes. Examples being informational microsites, internal-only informational sites and simple branded blogs.
B. **Custom site configuration with dedicated theme -** Sites in this category may require in-depth configuration and may require a heavily-customised theme - for example, to satisfy corporate branding requirements, accessibility policies, or simply to provide the most engaging user experience across a wide variety of devices and screen sizes.
C. **Custom module development -** Sites which require custom module development include any site which cannot be delivered solely using Drupal core and existing open source modules. The nature of the custom development is specific to site business requirements.
D. **Custom infrastructure or services -** Sites in this category require the use of additional infrastructure components or services to augment Drupal itself and the custom code already developed, for example advanced search and filtering capabilities for content, high-performance caching.
E. **Full-scale integration -** Projects in this category feature technical requirements which are substantially concerned with the integration between the web site and one or more existing or newly-developed systems.

The weighting of skills for a B2B project in E. Full-scale integration (above), would be over 70% for level 4 & 5 skills combined, reducing the involvement of lower cost developers to below 30%. This would normally make the cost of delivering the project much higher! However, the complexity involved in managing a complex environment introduces significantly greater levels of Platform.sh savings from full-stack branching and Continuous Integration (CI), thereby bringing the cost of delivery back down again.

Continuous Integration (CI) – e2e process automation

Implementations of CI have always existed, although less well understood than today and a lot more manual, slowly stepping the wider team through all the necessary activities to ensure code is checked, tested and works with all the technologies and services present in the live environment. What we do know with certainty though, is that the more we can automate this process end-to-end, and manage it with rules and triggers, the faster it becomes and the fewer people we need to oversee it.

There are over 50 common toolsets in the CI ecosystem today, making many aspects of the change management process more efficient, including versioning, tracking, configuration, testing and deployment. Although they are all steps in the right direction, very few of them are able to programmatically drive the hosted infrastructure itself, and only when this difficult step has been taken can you start to properly automate the process from end-to-end; from code-to-live.

Because Platform.sh is Git driven and offers an open API, it becomes a central cog in your CI machine, allowing your toolsets to drive change into it and solicit status information back out. And because Platform.sh enables on-demand branching of the full stack, plus snapshot back-ups, your other CI tools are able to programmatically create and destroy hundreds of perfectly consistent environments whenever their own internal processes requires it.

The alternative is a much larger and more permanent suite of static environments, all of which require careful and constant attention to prevent them drifting away from expected configuration states, which of course is the most common cause of problems.

Platform.sh therefore sits at the heart of your CI regime, and when fully automated supports the Agile process even further (see **Platform.sh – Unlocking Agile**).

How Platform.sh is changing our approach to CI	Savings and gains
"We have 7 Systems Administrators and DevOps, 3 of whom are now doing a lot more billable work" "We are selling a Continuous Delivery capability to retail now"	**Circa. €350k**

Offshore / remote working model

The value of Platform.sh to an organisation using off-shore and remote worker development teams is largely a measure of skillset downshift (see earlier section) and far better governance.

A number of major issues are immediately solved by using Platform.sh as the common development tool. For a start, cloned environments are provided on-demand, against a steady stream of new developers, and the coding of hundreds of features can now be massively parallelised.

Another area of concern is code quality, which is constantly being monitored and improved by our code checking & performance analysis tools. In addition, better management information on developer performance is provided through reporting of relative number of commits, successful merges, failed builds etc., allowing you to more accurately define KPI's and coding standards. Rotating new staff in to replace poor performers now becomes objective, quick and easy.

Platform.sh then facilitates automated testing within a better Continuous Integration (CI) regime, making code/features available for review and UAT (via hundreds of unique URLs) prior to sign-off and merging upstream. Allowing user communities to immediately review, discuss and approve changes with the development team by providing them a link to that page in the test system massively expedites this painful leg of the process.

So, combined with further 'skillset downshift' savings and the removal of even more infrastructure management costs, you can now bring to bear a much more efficient army of developers without the usual quality, testing, process and infrastructure management constraints. The overall risk profile associated with offshoring is significantly improved.

Estimated savings here are **10-15% infrastructure cost reduction** (including DevOps) and far superior assurance and governance processes.

Infrastructure savings

Most specialist hosting vendors provide standard sets of environments for a pre-defined development-test-live workflow process, and they are always on, ever-present and account for a high price point. These vendors typically have a complicated and therefore expensive process involved in making changes or adding hardware and services. So it makes sense for them to sell you a package far bigger than you actually need, and one which they don't have to resource up for, to keep fine-tuning it for you. You may be able to justify this outlay during the initial development phase, but once the site has been launched it still needs ongoing maintenance, emergency security patching and occasional updates and feature changes, and so you are required to keep all these environments in place.

But you shouldn't have to pay for that sort of service, and waste money on all that capacity when you're not using it. Platform.sh allows on-demand creation and destruction of environments that allows you to use what you need when you need it, and pay for it as you go. And because we change the development regime such that many feature branches can be created in parallel, you don't need to worry about scaling up to 20, 30 or even 100 environments at the same time.

This customer had 6 permanent environments in 11 countries plus their index site with 4 environments.

Hosting infrastructure – before and after	Savings and gains
Previous hosting vendor 6 in each of 11 countries + 4 for the index = 70 environments	840 monthly payments
With Platform.sh 2 permanent in 11 countries + 2 for the index = 24 environments + Average 3 temporary in 11 countries + 2 index for 4 months elapsed	[288 + 140 =] 428 monthly payments
SAVINGS	**€246,400 p.a**.

IMPACT OF PLATFORM.SH ON REVENUE AND SALES

We have a fast-growing number of retail customers deploying new sites and migrating existing ones onto Platform.sh across the globe now. Many of these organisations are running growing transaction based-businesses, and all of them have been able to introduce Continuous Delivery (CD) techniques based on the safety net Platform.sh gives them with guaranteed consistency between testing environments and live, along with seamless scaling for unforeseen resource spiking. This means they can make instant changes to sales campaigns during peak traffic and not worry about any unforeseen but related effect on system resources.

They're also making considerable savings in the development process itself, as well as what they are paying for the hosting infrastructure in general.

Preventing revenue loss

The automation functions we introduce have a huge impact on return-to-operations (RTO) metrics in production, as well as providing the business with instantly scaling live environments. The triple redundant architecture that is Platform Enterprise, combines the guarantee of continuous availability with a 99.99% uptime SLA that removes the need for expensive failover services. Importantly, there are no more outages, as spiking is handled through our non-disruptive scaling architecture.

Recent customer examples where Platform.sh was introduced to prevent outages recurring:

- The Facebook campaign that triggered a sales event upon reaching 60,000 'likes'. Traffic was unexpectedly heavy and there was no facility to seamlessly upscale the hosting service to handle the load, resulting in servers down, revenue loss and reputational damage.
- A similar unexpected traffic spike occurred when a new discount was introduced by the marketing team causing another outage. The effort associated with the manual re-provisioning of the extra capacity took a number of hours, resulting in further losses.

Quote from CEO, Wauwaa	Preventable losses
"We no longer worry about outages during sales campaigns"	€100,000 and reputational damage

Improving sales

Bringing forward a functional delivery, or quickly making changes to features in live can directly translate to additional sales, especially during a sales promotion where traffic analytics or A/B testing is telling you that modifying the online proposition will bear better results. Trouble is making changes to the live site during peak traffic is considered too risky.

However, the Platform.sh consistency concept largely eliminates the possibility of features being successfully tested in staging, and then failing in production. And so, decisions can now be made to take advantage of micro changes in your marketplace, quickly making changes to boost sales, and all when you most want to do this, during peak traffic periods!

Recent customer examples where Platform.sh has enabled revenue gains:

- Updates to promotional products during the opening hours of the sales campaign would have invalidated caching, and without Platform.sh the resulting surge in back-end database reads would have disallowed these last minute updates from marketing.
- Fraud proof discounting functionality that couldn't be delivered into live because the testing servers were booked out for higher priority work. Continuing instances of fraud and the inability to maximise promotional opportunities resulted in significant revenue loss and lost time from management addressing fraud related issues.

Many of our retail customers are telling us that through holiday periods and promotions (Black Friday, Cyber Monday and December 26th) the reduced risk of making changes into live combined with Platform.sh's ability to seamlessly upscale around unexpected traffic peaks have resulted in significant revenue gains.

C level insights	Revenue gains
"several changes during the day that we wouldn't otherwise have been able to make. A great day"	About €30,000 today
"The last time this happened we couldn't reverse the discount codes"	Saved us €85,000