

Drupal Support and Maintenance services

Business case, Team composition, SLA guarantee, Reporting formats

#	ver	Description	Author	Date
1	0.1	Draft document	Rahul Dewan	27 Apr 2011

“Content Management Specialists”

Table of Contents

1	Case for Support and Maintenance services	3
1.1	Why is support needed?.....	3
1.2	Does open source software require support and maintenance.....	3
1.2.1	Not happy with current Drupal vendor? Choose another one!.....	3
1.2.2	“How many “bugs” did you write”?.....	3
1.2.3	“We need new features too”.....	4
2	Internal support or external support – which way to go?	5
2.1	Team Composition.....	5
2.1.1	Core Team.....	5
2.1.2	Fringe Support.....	5
2.1.3	An adequate team.....	6
2.2	Are you a software company?.....	6
3	Reducing the TCO	7
3.1	Srijan's unique service offering.....	7
4	Engagement Model	8
4.1	Insurance/Team availability.....	8
4.2	Preventive maintenance (vs reactive maintenance) activities.....	8
4.3	Bug resolution.....	8
4.4	New Feature requests.....	8
5	Processes	10
5.1	Development and Staging environments.....	10
5.2	Process of a maintenance release.....	10
5.3	Tools used.....	11
5.3.1	JIRA Studio.....	11
5.3.2	Basecamp	11
5.4	Reporting.....	11
5.5	SLA.....	12
6	References	14

1 Case for Support and Maintenance services

1.1 Why is support needed?

Between the years 2006 – 2010 roughly 50% of the software activity globally was in maintenance of existing software implemented at corporations¹. Clearly organizations require services for *bug-resolution*, *internal usage training*, and *enhancements* of the applications implemented.

There has been a lot of backlash² against the software industry in earning disproportionately high margins³ from support services, and the vendor-lock-in.

1.2 Does open source software require support and maintenance

Open source software has been projected as the solution against vendor-lock-in and recurring high support costs, resulting in high TCO of the implemented software.

This is correct! There is not one *SAP* or *Microsoft* offering support for the product you have bought and implemented.

1.2.1 Not happy with current Drupal vendor? Choose another one!

Organizations implementing *open source software*, specifically Drupal, to drive their intranet applications or their externally facing web applications, have a *free choice* to choose between a range of vendors available globally for *support and maintenance*.

However, the notion that open source software DOES NOT require support is INACCURATE. On the contrary the key reason why open source software *lacks enterprise penetration globally* is attributed to the lack of support available, and the high costs of hiring internal resources.

Choosing another Drupal vendor will have a small initiation cost, while the vendor does a setup of the system and its team gains business and application knowledge, apart from reviewing the code and implementation.

This cost is however, negligible if compared to the vendor-lock-in situations in case of proprietary software.

1.2.2 “How many “bugs” did you write”?

Introducing bugs in the process of software of development is an accepted phenomenon. While open source software is known to be *more secure* than proprietary software, it is also prone to more software bugs introduced as developers with varying skills (and not working within structured development environments) can contribute modules.

It is towards this that Drupal (an open source software for *enterprise web content management* solutions) recently introduced training for *community code reviewers*⁴ to help them become better

code reviewers, and thus ensure better quality of code that is released in the community.

The maintainers of various Drupal modules constantly release code-patches with new features and even security updates.

Someone needs to be constantly engaged with Drupal, and specifically in these code enhancements, with well-defined processes, monitoring and reporting mechanisms in place, in order to be pro-active about applying these patches on your live/production deployments.

So, YES, your software requires bug-resolution support even after it has been tested and is in a *production environment*. Whether you do this internally or through a Drupal vendor is a choice based on economics that you would need to make.

1.2.3 “We need new features too”

If your business is growing, it is surely evolving its services for its clients as well. It is probably imperative for you to keep pace with these growing communication and client-engagement (internal or/and external) needs. And your internal and external Drupal applications need to reflect these changes.

This is the other important reason for engaging in a Drupal software maintenance contract.

“Content Management Specialists”

2 Internal support or external support – which way to go?

2.1 Team Composition

A typical team composition required to maintain a Drupal applications is as follows:

2.1.1 Core Team

1. **Project Manager** : A set of developers cannot maintain applications. There needs to be planning, test plan management, execution oversight, documentation, resource planning and monitoring, and effective reporting. These are skills acquired with years of experience in being part of and managing software development teams.
2. **Team Lead** : A senior Drupal developer having work experience across CSS/theming and module development to ensure the larger goals of an implementation are met – some of these are: a) following coding standards, b) validating the right modules to be used for specific business cases, c) ensuring timely deliveries.
3. **Drupal developer(s)** : Someone who has been trained in writing good Drupal code
4. **Drupal themer(s)** : An expert in CSS and specifically Drupal theming
5. **Quality Assurance Engineer(s)** : Someone who is trained in writing manual and automated test cases (in tools such as Selenium), and in documentation of specifications and test cases.

2.1.2 Fringe Support

1. **Business Analyst** : who can understand and document the business needs and help translate these into a measurable set of feature requests; often the same resource can double up as a client-interface manager as well.
2. **Technical Architect** : Someone who has experience designing and architecting applications across technologies, with several years doing so with Drupal applications. The key role this person performs, in application maintenance scenario, is that of a *mentor* and *troubleshooter*.
3. **Deployment Manager** : There is a need to have extensive and carefully controlled deployment processes in place, specifically in cases of mission-critical websites involving *payment transactions* or high-uptime scenarios.

2.1.3 An adequate team

This we believe constitutes an adequately staffed Drupal support team.

With any of these functions missing, the team would nearly always remain wanting on several accounts such as documentation, quality control, deployment and release processes and overall quality of the application.

2.2 Are you a software company?

In your choice between having an adequately staffed, and yet an internal team, to support your Drupal applications is whether you are *in the business of software development*, or would you rather focus on your own business.

Further, are you willing to deal with the *high cost of attrition*, and *knowledge transfer* issues within the development team?

“Content Management Specialists”

3 Reducing the TCO

The reasons for an industry backlash² as cited earlier, is primarily overarching and often dedicated-resources based support contracts.

Most often, an adequately staffed dedicated team is not occupied 100% of the time available in maintenance activities. Maintenance is often a low %age consumption of time, in varying capacities across these resources.

3.1 Srijan's unique service offering

Hence, Srijan has come up with unique offering of blocking *billable hours* across an adequately staffed support and maintenance team.

This offers significant advantages to our clients. They get the benefit of availability of all desired *roles* and *processes* to maintain a software development lifecycle, yet *at a fraction of the cost* of internally or externally positioning such a team.

4 Engagement Model

4.1 Insurance/Team availability

As part of our cost structure, Srijan requires a certain minimum number of hours each month in to ensure availability of a team who can respond to your support requests. This time ensures that we deal with challenges of *knowledge retention* of your business application within our team, even in cases of attrition and other similar challenges of the software development industry.

There is an setup/induction fee for Drupal applications not built by Srijan.

4.2 Preventive maintenance (vs reactive maintenance) activities

The key reason for staffing the support and maintenance team adequately with all the roles defined earlier is that there can be preventive maintenance (as opposed to reacting to client-driven bug-resolution requests or even emergency escalations).

1. Identifying and applying security/enhancement patches for *contributed Drupal modules*
2. Identifying and applying security/enhancement patches for *core Drupal*
3. Maintaining custom-modules written by us during development (if the application/website has been developed by us)
4. Code clean-up (and *code review* and *code documentation*)
5. Test Cases Documentation
6. Release documentation

4.3 Bug resolution

Bugs resolution remains an integral part of maintenance. Bugs are most often identified by clients during the use of the Drupal systems deployed.

4.4 New Feature requests

New feature requests can be client-driven; and often recommended to the client by our internal team (Business Analyst function).

1. Review meetings, discussions, prioritization and planning
2. Feasibility and impact analysis (on the overall architecture of the application)
3. Technical design/Architecture documentation
4. Development

5. Coding
6. Quality assurance
7. Deployment
8. Documentation, including release document, and user manual

“Content Management Specialists”

5 Processes

5.1 Development and Staging environments

Srijan sets up a development and a staging environment for all Drupal applications we maintain.

The *development environment* is where our developers collaborate to write new features or resolve bugs, while the *staging environment* is a replica of the live/production database, so as to offer a testing environment for our internal Quality Assurance team as well as for the client to verify and approve maintenance releases.

5.2 Process of a maintenance release

There is a clear, well-defined process that Srijan's maintenance team follows during any maintenance release. An *exhaustive* list of activities that a maintenance release requires is defined here:

1. Create tasks and sub-tasks for bugs/features
2. Document test cases
3. Checkout most recent codebase from SVN repository
4. Create SVN Branch for new release
5. Code/Theme on issues created
6. Write automated test cases
7. Deploy on “Dev” server
8. Merge branch with SVN Trunk; Create new Release Tag
9. Established *continuous integration* process with each release of *performance* and *automated test case management using Hudson/Jenkins* process
10. Deploy on “Staging” server
11. Handover to Quality Assurance (QA) team for manual and automated testing
12. Handover to client for *User Acceptance Testing (UAT)*, if required
13. Deploy on “Production” server
14. Report to client
15. Training the client in usage of new features/enhancements
16. Updating project documentation and user-manuals (if any) based on changes made

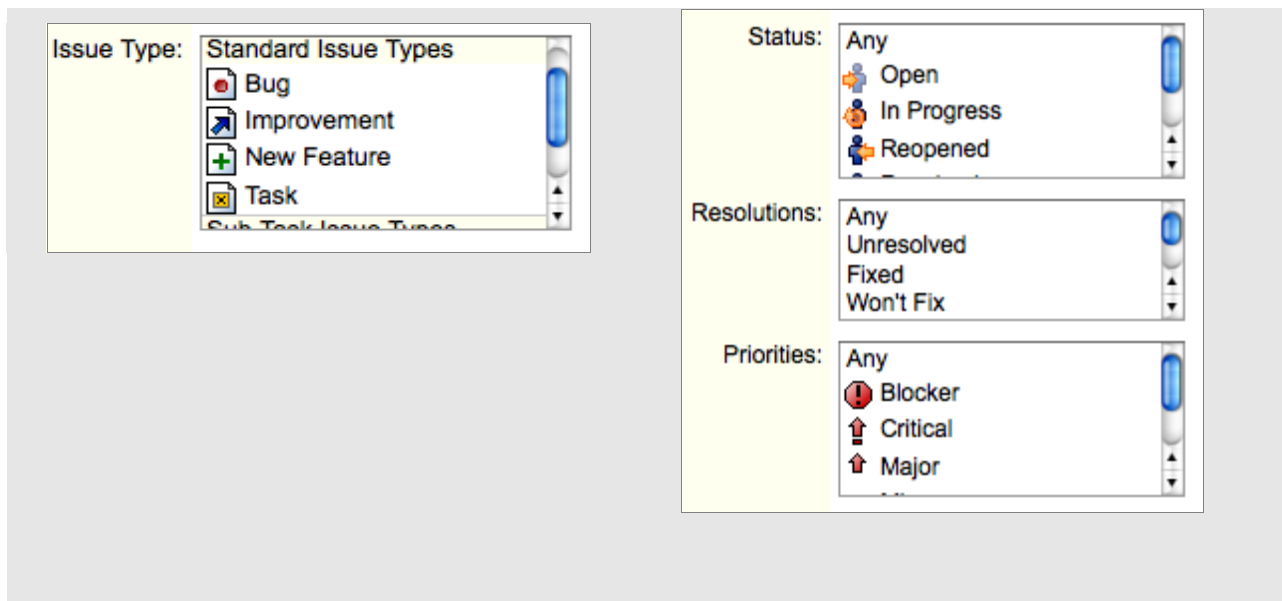
5.3 Tools used

5.3.1 JIRA Studio

All 'issues' (features and bug-fixes) would require reporting through *JIRA Studio* the *Issue Tracking* system we use, implemented at <http://apps.srijan.in/tracker/secure/Dashboard.jspa>.

Issues can be tracked with varying levels of Types, and be given various attributes, such as level of priority (Blocker, Critical, Major, etc.).

Other than *Issue Tracking*, we use *JIRA Studio* for *code reviews*, *code version management*, as our *code repository*, and *documentation wiki*.



5.3.2 Basecamp

Basecamp is an additional tool we use to increase collaborative communication with our clients.

5.4 Reporting

The heads that would be covered in the reporting formats is as follows.

1. Period, Project, Client
2. PM, SPOCK name
3. Dev, QA team-leads, resources count for the month
4. Outline of tasks accomplished in the current months (and previous months)
5. Hours consumed

Description of the “support and maintenance” model

6. Tasks pending from previous month, if any, with reasons
7. Tasks planned for next month
8. Issues, Dependencies (Technical -- status of any serious or show-stopper issues resolved, with reason for occurrence, etc)
9. Infra, resource and other issues and dependencies, if any

5.5 SLA

A Service Level Agreement (SLA) needs to be defined on a case-to-case basis with our clients. This would then determine the blocked hours and the price engagement on the maintenance contract.

A checklist for arriving at the cost of engagement is defined below:

#	Service	In scope	Estimated effort per month / per quarter	Comments
1	Availability of a skilled Drupal team	✓	__ hours	Minimum hours required for meeting SLA agreements.
2	Fortnightly calls with client to discuss next set of tasks and report <i>accomplishments</i>			
3	Internal planning and management tasks for improving the product			
4	Executing work agreements made with the client for adding new features OR working on improvement suggestions prepared by our team.	✓	__ hours	Minimum number of hours allocated for supporting the application.
5	Website hosted on Srijan's server environment?	x/✓		
	– Regular server maintenance activities	x/✓	__ hours	
	– Mission critical website? Requires immediate response time in case of downtime?	x/✓	__ hours	Support for mission-critical websites is currently limited to India day-time support; India night-time emergency support is offered on a best-effort basis.
	– Cost of Server environment	x/✓	_____	Costs of hosting if server is hosted on Srijan's environment

“Content Management Specialists”

#	Service	In scope	Estimated effort per month / per quarter	Comments
6	Website hosted externally?	x/√		
	– Mission critical website? Requires immediate response time in case of downtime?	x/√	__ hours	Support for mission-critical websites is currently limited to India day-time support; India night-time emergency support is offered on a best-effort basis.
7	Content update activities and support	x/√	__ hours	Hours allocated for content updates on the website, if required
8	Deployment activities	√	__ hours	Hours allocated for deployment related activities

“Content Management Specialists”

6 References

1. <http://www.compaid.com/caiinternet/ezine/capersjones-maintenance.pdf>
2. <http://www.zdnet.com/blog/btl/why-theres-a-software-support-and-maintenance-revolution-underway/9717>
3. http://dealarchitect.typepad.com/deal_architect/2005/10/killing_the_goi.html
4. <http://groups.drupal.org/code-review>
5. <http://www.zdnet.com/blog/howlett/a-fresh-model-for-software-maintenance/459>