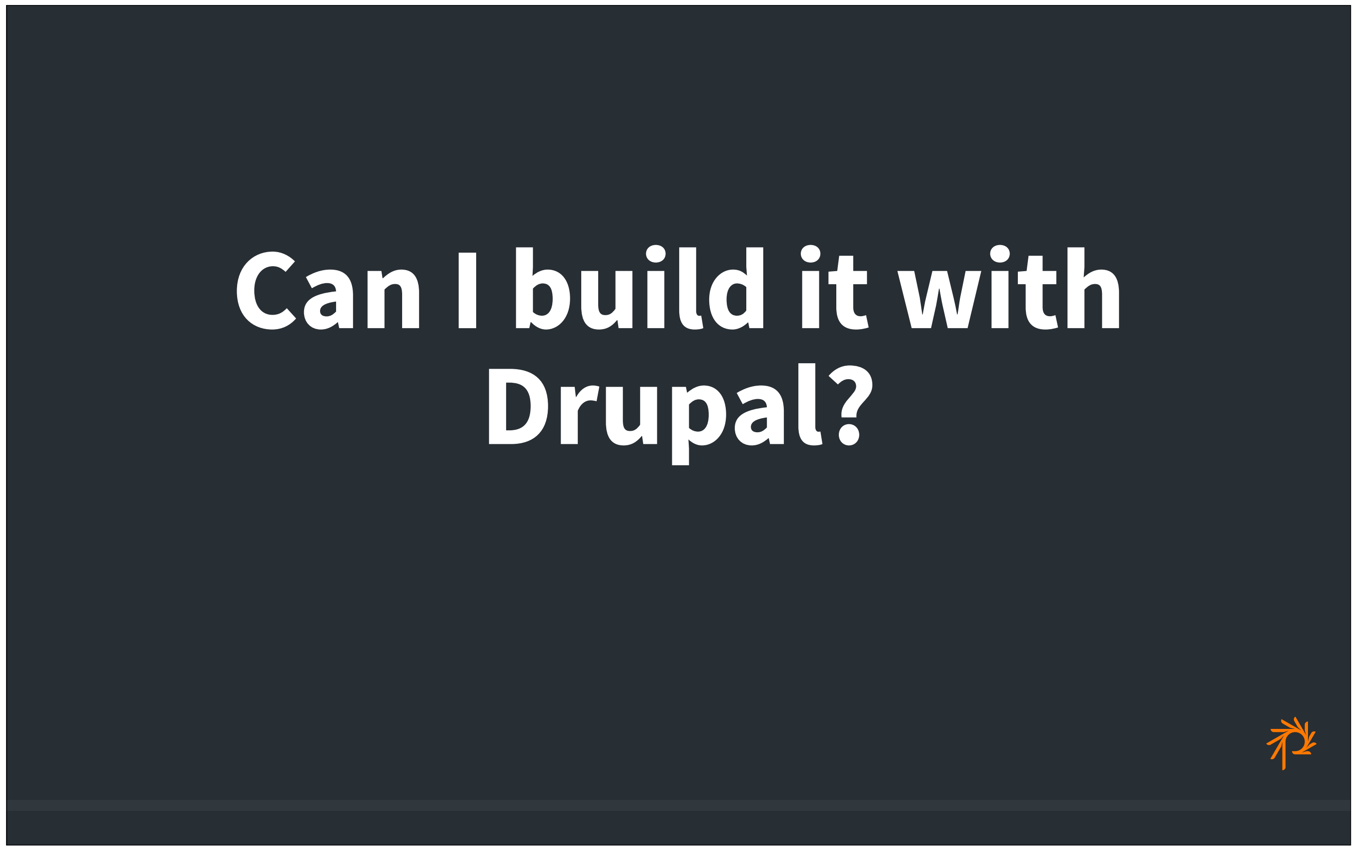
The ‘YES’, ‘NO’, and ‘MAYBE’ of

‘Can we build that with Drupal?’

Joshua Lieb, Phase2

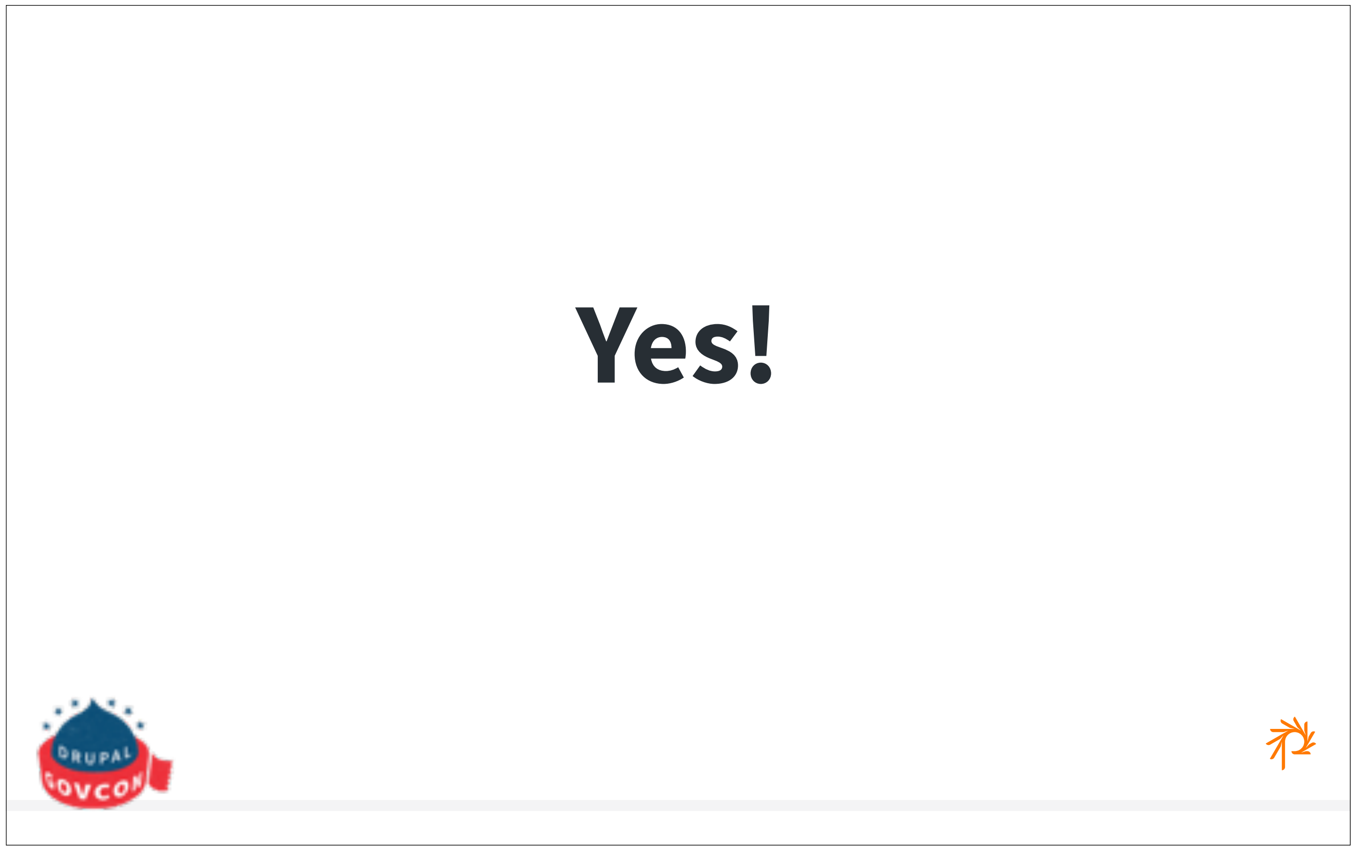
July 23, 2015
Drupal GovCon 2015



Can I build it with Drupal?

Drupal has come a long way in the last fifteen years – I don't know what the first government website to use it was, but the governmental and public sectors were early and enthusiastic early adopters, from Howard Dean's 2003 campaign through the launch of the White House site in 2009 and through to today.

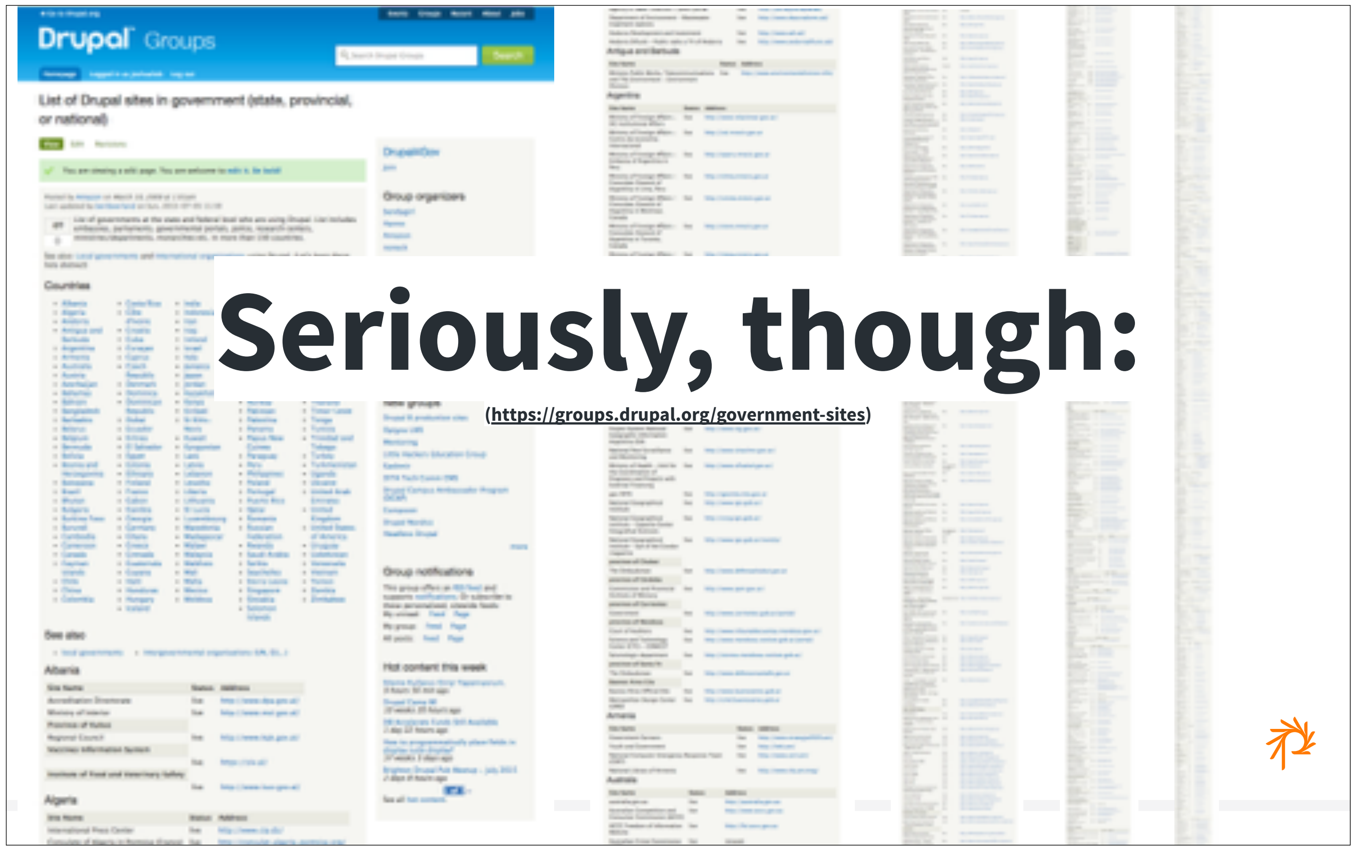
At Phase2 we've been involved with Drupal and government both for a long time, we've been part of this process, and we've seen Drupal become a default option – if it's not already the choice, then there's always the question: Can I build it with Drupal?



Yes!

And that concludes today’s talk.

Enjoy your lunch! It will be ready in 45 minutes.


Drupal Groups

List of Drupal sites in government (state, provincial, or national)

Seriously, though:

(https://groups.drupal.org/government-sites)

Drupal has already proven that it is flexible enough, secure enough, and extensible enough to be used for a wide array of projects.

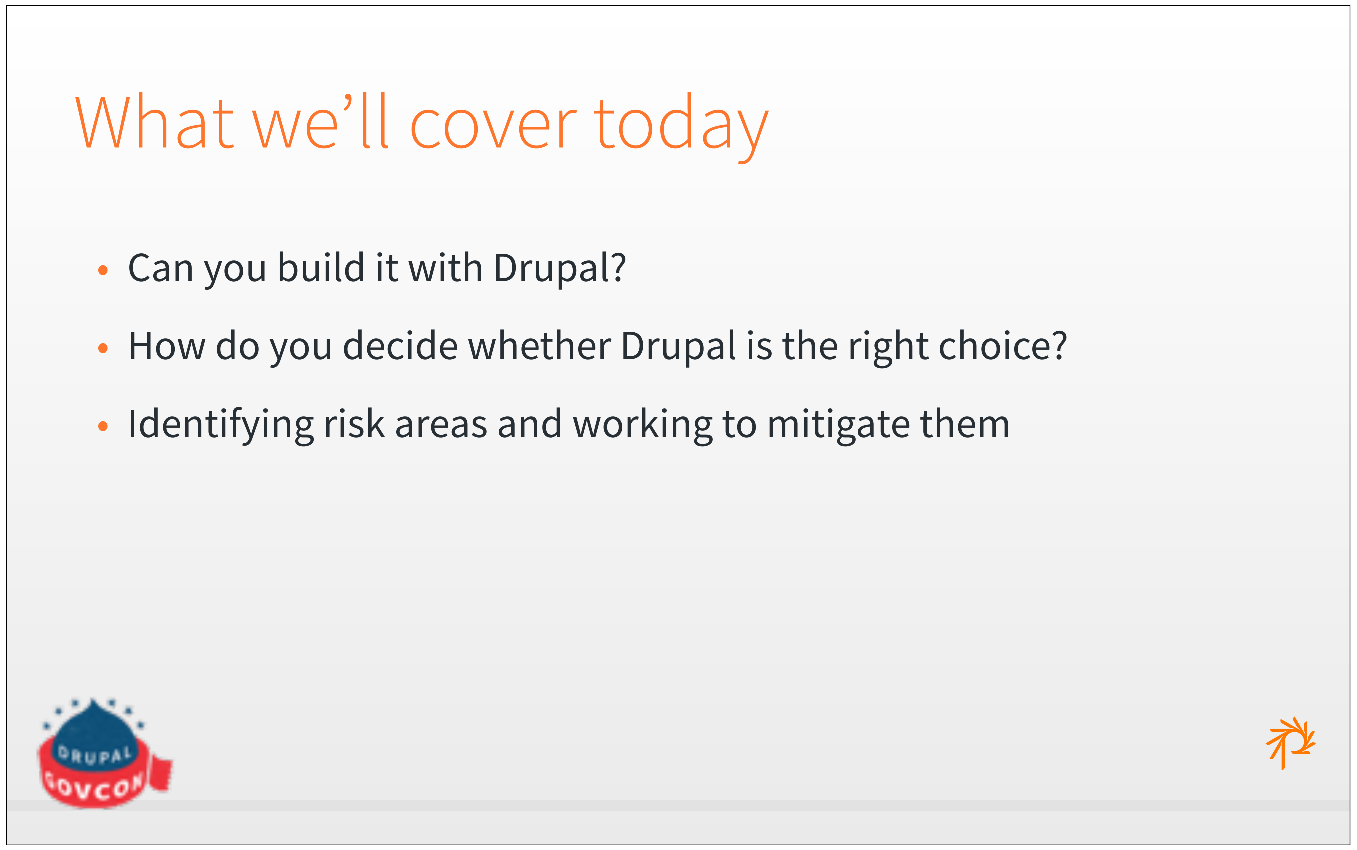
- High-traffic, high-security, high-profile sites like The **White House** public website and the **We the People** petitions site
- Multisite platforms used to manage networks of diverse sites for departments and agencies at the state and local government level
- Templated site-building platforms for institutions like the House of Representatives
- Collaborative NGO platforms for managing international teams of crisis managers

The page your seeing here is the ‘government sites’ list on Drupal Groups. I have no idea how accurate it is but taking a screenshot of this page shut down Chrome for like 5 minutes while it compiled.

Can you build a curriculum management system with Drupal? Sure (namecheck?)! How about a CRM? Check out (namecheck)! Collaboration and project management portal? OpenAtrium! A data management platform? A publishing system? Document management?

If there’s not a distribution, you can build your own. You can extend Drupal to integrate with third-party libraries, use the web services APIs to integrate with external datasources, and bridge the gap to external systems like Alfresco.

Can I use Drupal with Sharepoint? Sure! There’s a module for that!



What we'll cover today

- Can you build it with Drupal?
- How do you decide whether Drupal is the right choice?
- Identifying risk areas and working to mitigate them

This discussion is largely going to focus on what happens BEFORE the project gets started – deciding on a platform, settling on an architecture and development plan.

After that, obviously you still have work to do to make sure that the project goes well, but at this point you've gone past our core question 'Can I build this with Drupal?'

There's plenty of excellent sessions if you want to tighten your game in other areas. It's unavoidable that some of what I'm going to talk about is related to general best practices for running projects, but I'm going to try to focus on looking at how to make sure that your plans make the best use of Drupal's strengths, and avoids some common downfalls .

Joshua Lieb

Digital Strategist, Phase2

Email: jlieb@phase2technology.com

Bio: https://www.phase2technology.com/joshua-lieb/

d.o: https://www.drupal.org/u/joshualieb

I'm Joshua Lieb, I'm a digital strategist with Phase2, I've been with Phase2 for going on six years, and been here in DC since 2001. My job, as a strategist, basically, is to work with a client to figure out what they need, work with our engineers to figure out how to build it, and then make sure that what we build is faithful to what the client needs.

I've been working with Drupal for ten years, and in fact I first encountered it through a website that I was building for a governmental client. I had been working at a governmental association for a number of years building out their homebrew PHP CMS, and when I left that position I had some ongoing contract work with a related organization and needed a platform to build their site out on.

The CMS I had written had become increasingly hard to maintain, extend, document, debug – I was one guy. But here, here was an open source project that already did many of the things I previously had to build myself – user management, flexible content types, abstracted theming layers, tagging, access controlled content…and there was this huge community supporting it.

I didn't need to build everything!

It more than met the basic requirements of what I needed it to do for my project. And clearly it was built to be extended, and I figured that I could just (you know) customize it as needed to fit the rest of the requirements.

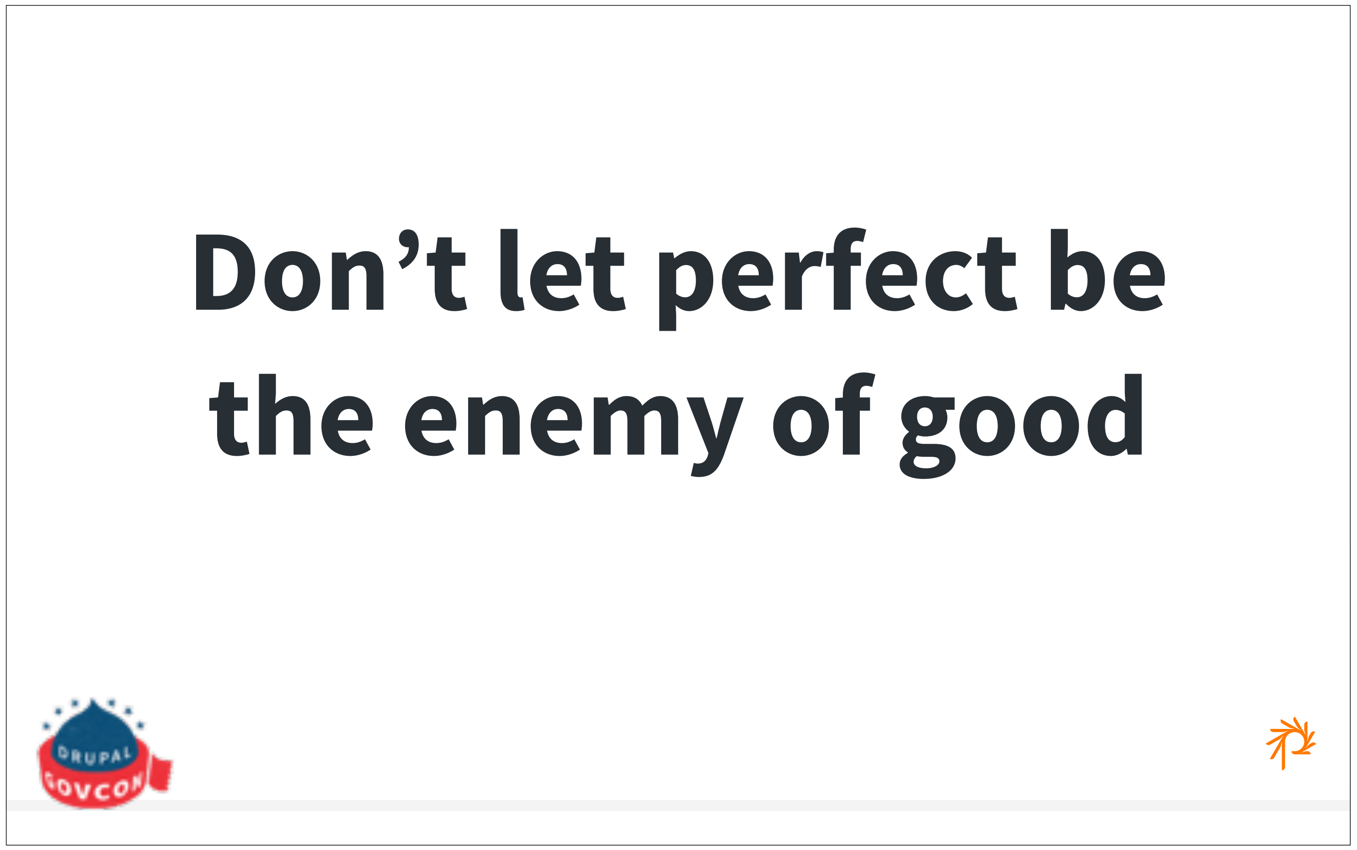
My project did not go well.

As I got deeper into development, I found that although many of the available modules met requirements at a basic functional level, they didn't meet them exactly, and when I tried to get different modules to work together there were conflicts and functional gaps, and I soon discovered that I had greatly overestimated my ability to code my way through these problems.

My project was a failure, I'll come right out and admit it. It dragged on and on until I had to admit to the client that we needed to start over.

Failure can take a number of forms, but I think we can generalize and typify these modes of failure:

- Some combination of the classic triad – over budget, broken schedule, bad software
- In some cases, you just don't launch your site or product
- And in some cases, you do launch, and that can be worse, especially in the government domain



Don’t let perfect be the enemy of good

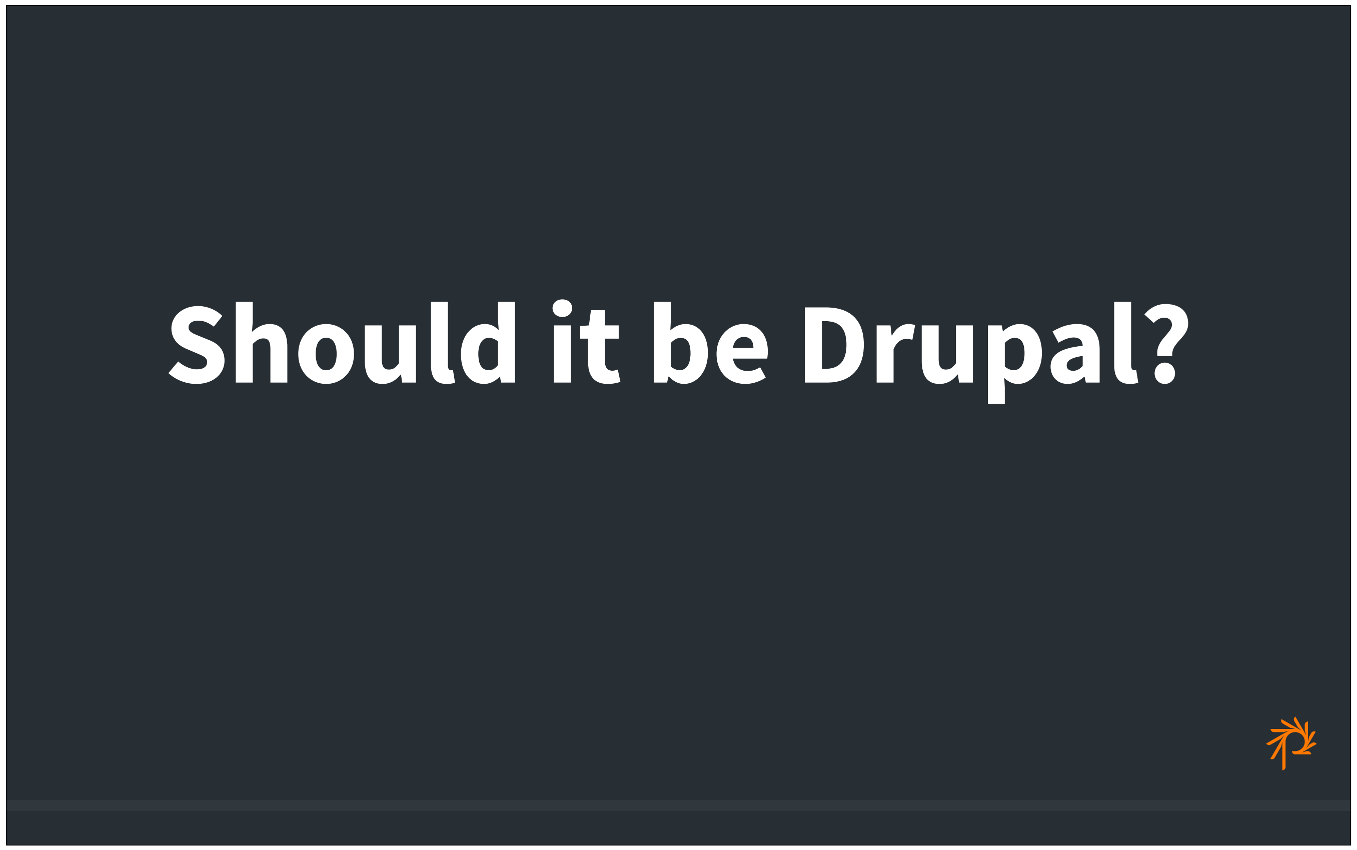
The problem wasn’t with Drupal. I had looked at the breadth of the functionality available in the contributed module library, at the robust developer community, at the extensibility of the underlying framework, and I had assumed that **I could make Drupal do what I wanted it to do** –

I should have started WITH Drupal, figured out what how I could work with its core strengths, and adjusted my requirements. If you are going to use Drupal, you need to understand what it’s strengths are, what its architecture demands, and the unique effects that it has on planning and running a project.

If you want to succeed with Drupal, you have to be willing to change. You have to be willing to adapt your internal processes, your relationship to your software, the way you think about your mission.

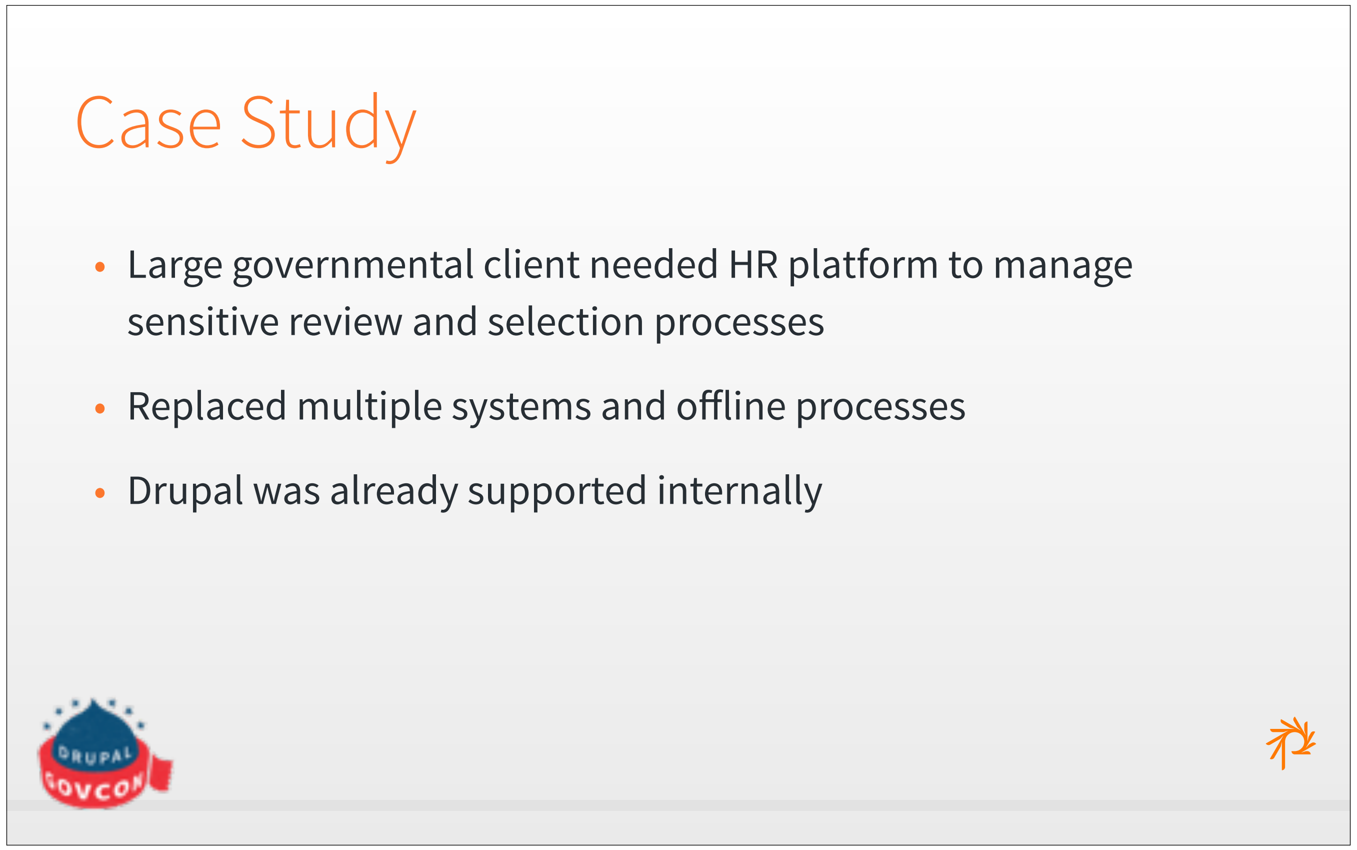
But you should be changing, anyway – organizations need to evolve to meet new challenges, and frankly one of the great strengths of Drupal is that it can evolve with you.

But sometimes you have to take baby steps. Sometimes you can’t control all of the externalities, so the next step, after deciding that ‘Yes, we CAN build it with Drupal’ is asking the question



Should it be Drupal?

This is distinctly different than ‘Can it be Drupal?’



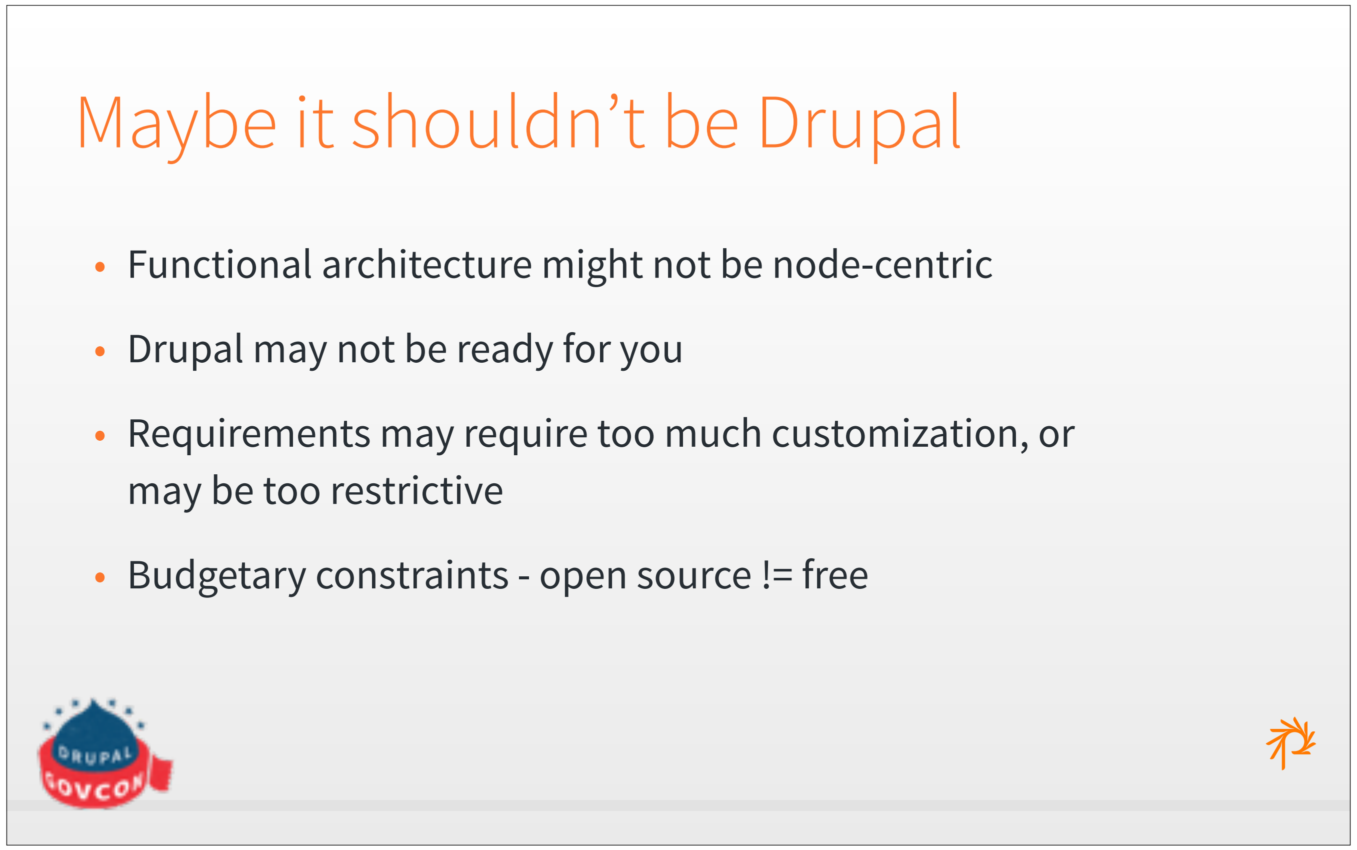
Case Study

- Large governmental client needed HR platform to manage sensitive review and selection processes
- Replaced multiple systems and offline processes
- Drupal was already supported internally

Let’s look at a case study.

This project was undertaken for a large government client that needed an HR application that would be used to manage the review, selection, and vetting process for employees. This replaced a set of legacy systems that were written in a couple of different languages, and took online several complex review processes that previously had been done offline using Excel spreadsheets.

We knew that Drupal was favored – we had worked with this client on several large-scale Drupal projects, and there was institutional support already in place. There were some red flags, though, that popped up immediately, and when the project was first brought to us we proposed that we start with a distinct discovery phase to assess the technical architecture to see whether our answer to ‘Can you build it with Drupal’ would end up being that we ’should build it with Drupal’.



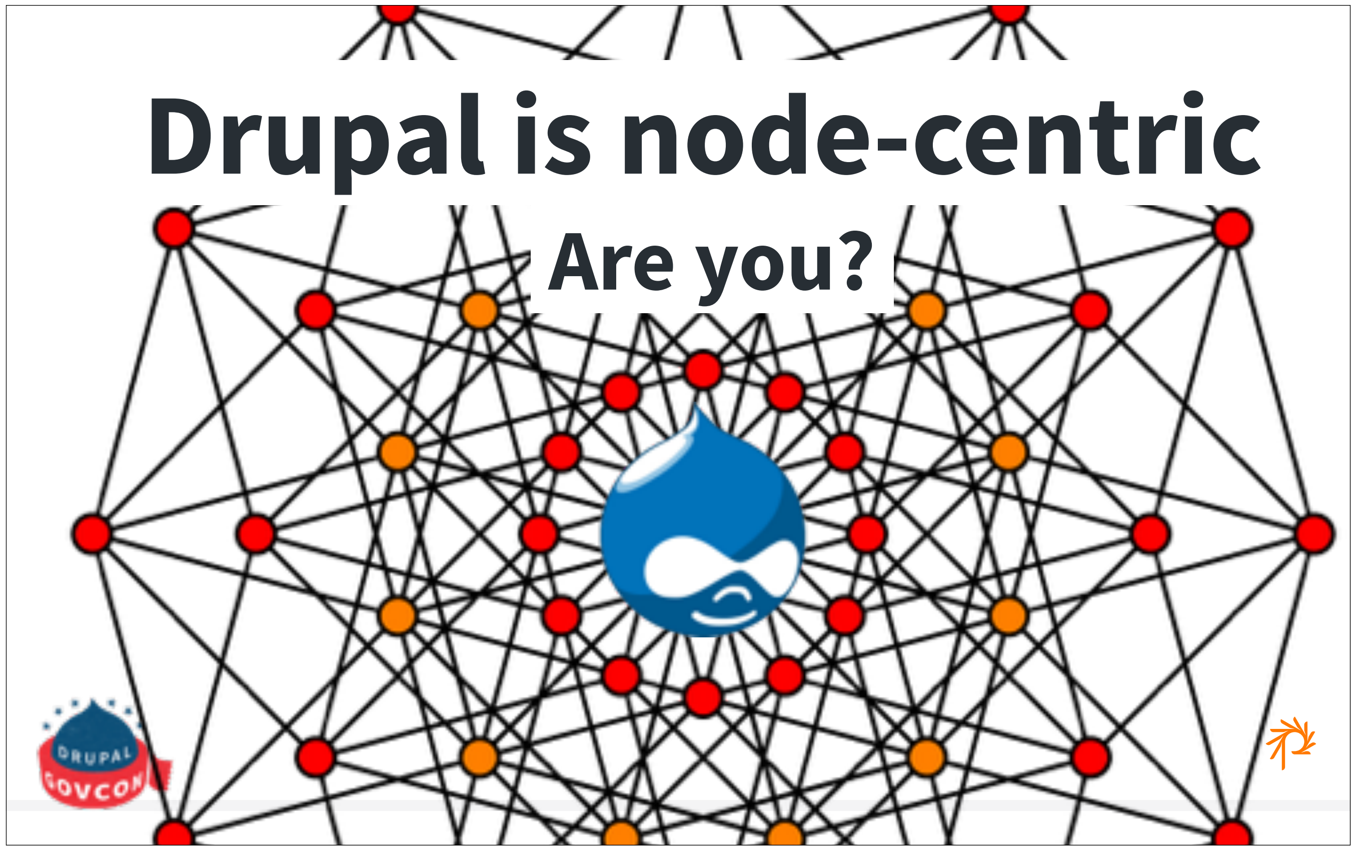
Maybe it shouldn’t be Drupal

- Functional architecture might not be node-centric
- Drupal may not be ready for you
- Requirements may require too much customization, or may be too restrictive
- Budgetary constraints - open source != free

There are many reasons why you might choose not to (high level)

- Functional architecture might not be node-centric
- Drupal may not be ready for you
- Requirements may require too much customization, or may be too restrictive
- Budgetary constraints – open source != free
- Institutional or contractual constraints

Of course, this is not to say that should any of these conditions apply to your project you would be precluded from using Drupal – just that these introduce an element of risk that might, alone or in combination with other risk factors, cause you to either rethink using Drupal, or rethink HOW you use Drupal.

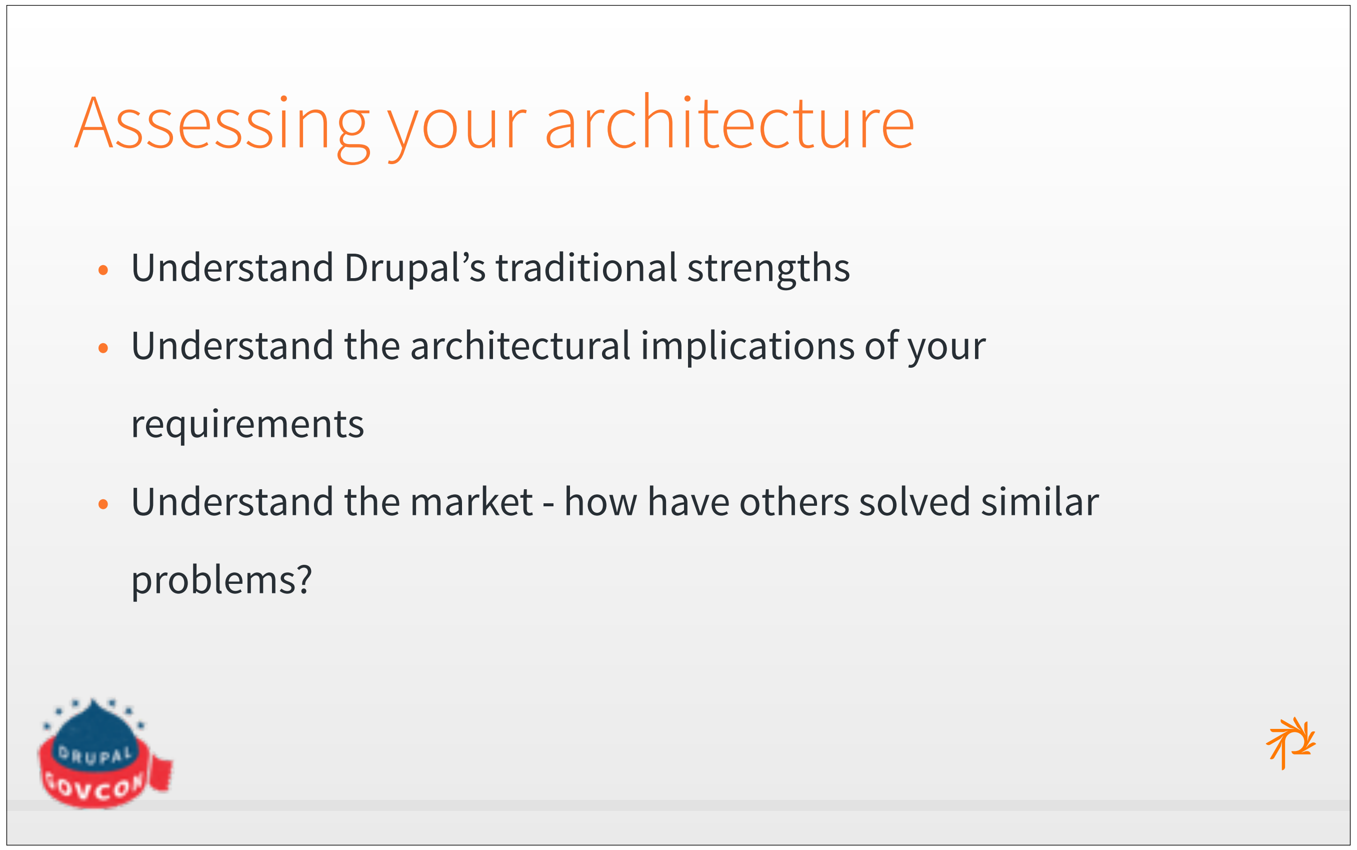


Drupal is node-centric

Are you?

When I am describing how the modular framework of Drupal is structured, I'll often use a fractal hub-and-spoke metaphor. If you think of Drupal core as the central node in the larger universe of contributed modules, and draw linkages to the most common and widely used 'near core' modules, and then linkages to modules related to those modules, you'll start to see a network emerge – core at the center, orbited by Views and Context and Features and similar 'near core' modules, and concentric layers of increasingly less-used modules.

This architecture is node-centric – by this I mean, that the central architectural assumption that is at the core of Drupal is that the node is the base unit of content. By and large, contributed modules have at this assumption at their core – they extend the node, they operate on the node, they display the node. If you find an interesting module that does something that you think would be core to your goals, it's likely going to be working at the level of the node – entities, users, taxonomy terms, and another objects in the system can't leverage this code in the same way.



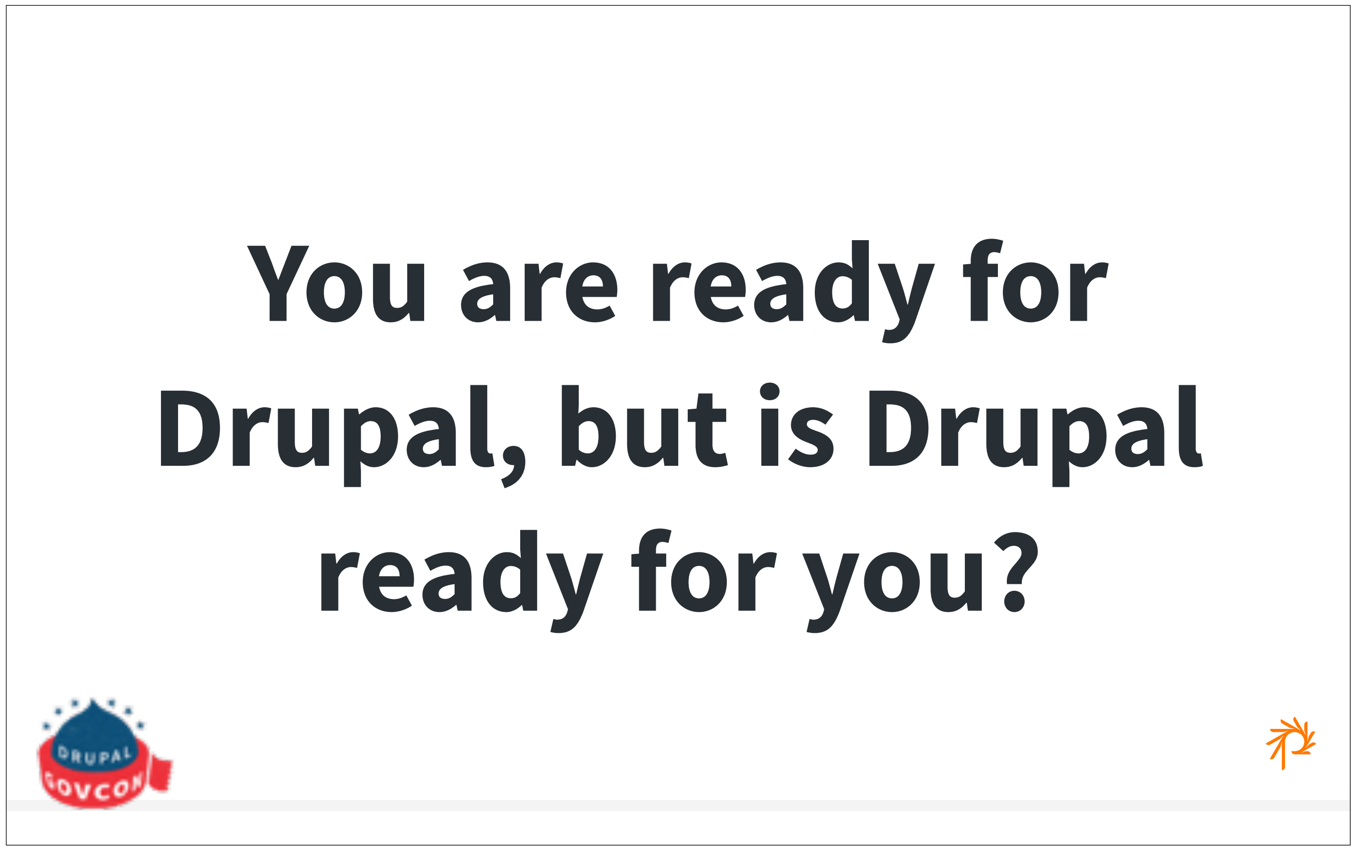
Assessing your architecture

- Understand Drupal's traditional strengths
- Understand the architectural implications of your requirements
- Understand the market - how have others solved similar problems?

In the case study of the HR application, we used entities to represent each candidate due to the complexity of the data model.

However, this meant that we couldn't use existing workflow modules like Workbench since those were node-centric. We had to write custom code to do very basic things for which there were existing modules, had we been able to use node-based content types.

So what can you do? Compare your project or product against the marketplace – if there are good examples of Drupal sites with similar functional profiles, sites that are already doing what you want your site to do, it's a good bet that architecturally speaking, your site won't be breaking new ground (and this is a good thing). If there's not many examples, then you need to make sure you understand the architectural implications of your project.



You are ready for Drupal, but is Drupal ready for you?

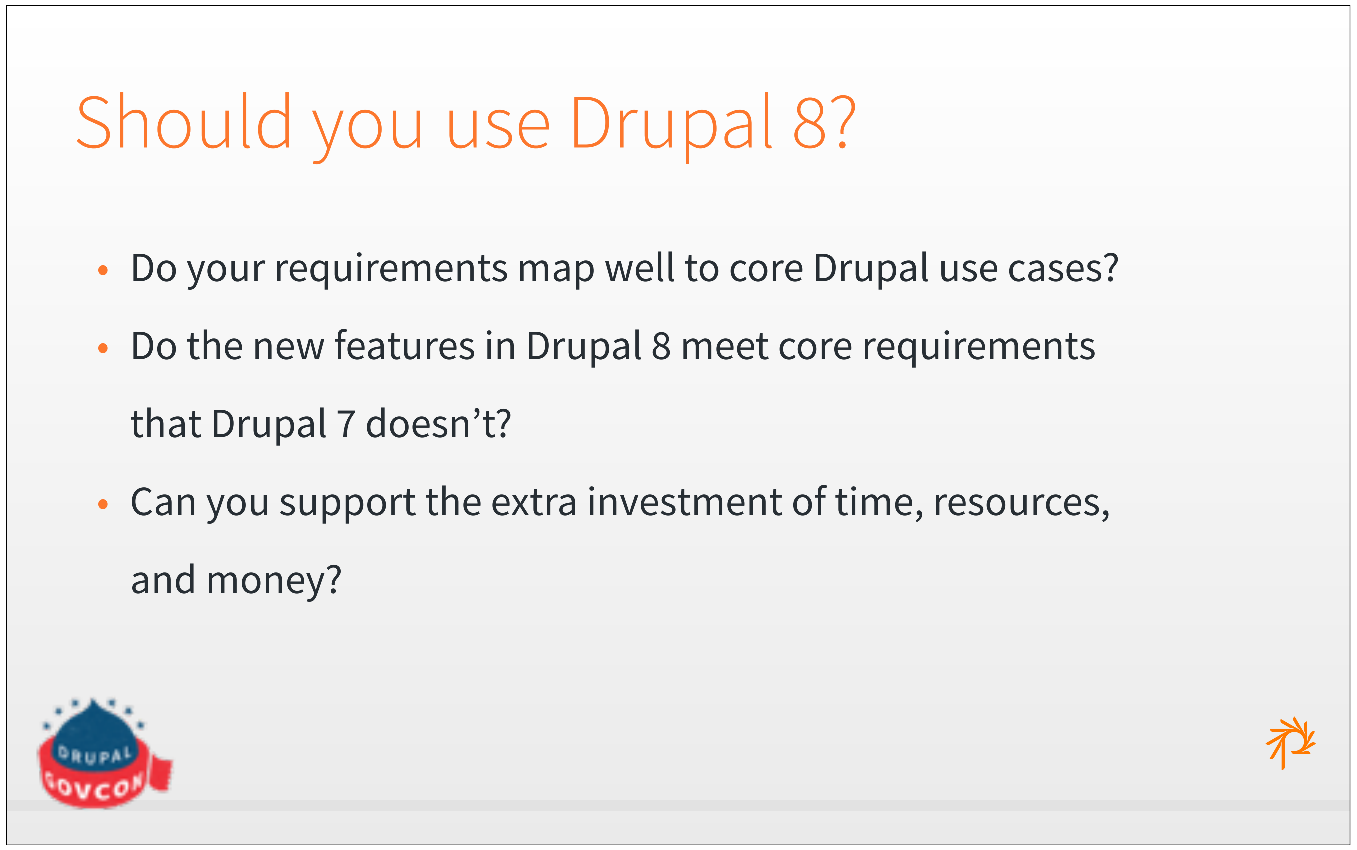
A historical note on our case study: this was undertaken very early in the Drupal 7 release cycle and was one of the first D7 sites we built. In fact, when we started working out the architecture of the HR application, it became clear to us that for various reasons nodes would not be suitable, and the only way we could proceed with Drupal would be to use entities, which were new to D7.

This presented some challenges.

To return to the hub-and-spoke model, the further down the spokes you go, the less stable and mature the intersections between modules gets – and the longer the distribution is out there, the more code is contributed to knit the downstream modules together.

For example, there are any number of field types available – for addresses and locations, geo co-ordinates, dates, media files of different types – and because the Fields and Views modules and APIs are so mature, they work well together. Add a field of this new type, and you can query on it using Views.

Working as we did with Entities very early on, the integrations with Views needed a lot of work and extension to get them working in the way that we needed – work that wouldn't have been necessary had we done the work years two years later.



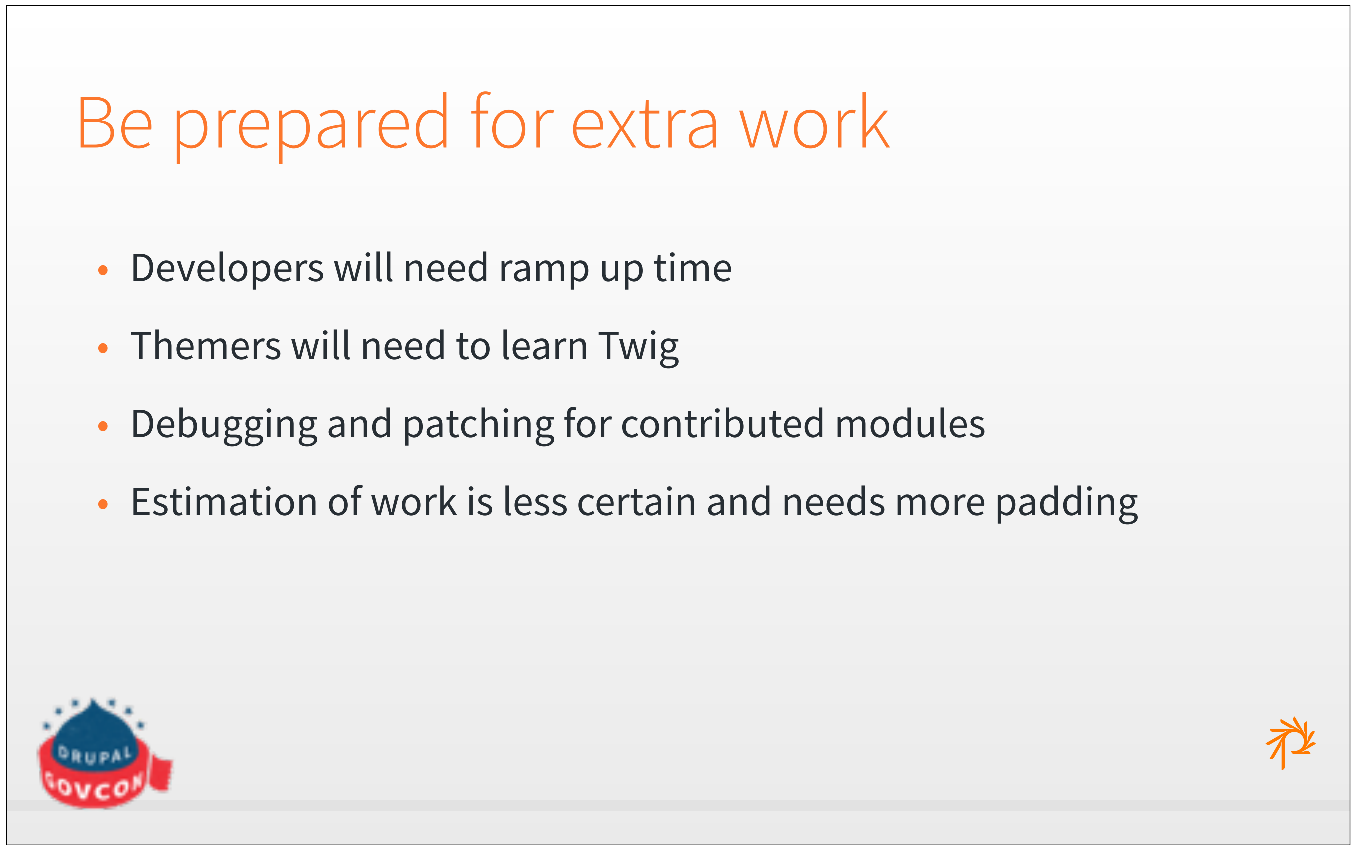
Should you use Drupal 8?

- Do your requirements map well to core Drupal use cases?
- Do the new features in Drupal 8 meet core requirements that Drupal 7 doesn't?
- Can you support the extra investment of time, resources, and money?

So, should you use Drupal 8?

This is particularly salient for those of us who are having to advise clients and plan projects with Drupal 8 in mind – just as Drupal 7 offered new features and capabilities that opened up entirely new use cases, Drupal 8 will do the same, and it will be tempting to jump forward. But – we need to be cautious, and think about whether our project is right. As a general rule, Drupal 8 makes particular sense for two types of projects:

- Where the core objectives and architecture hew closely to Drupal's core strengths, where the real knotty issues were already ironed out over the last few versions.
- Where the core objectives of the site depend on the new features offered in D8 – for example, the emphasis on separating the presentation layer and the provision of a complex web services framework would be a good fit for a platform requiring diverse publication paths
- In both of these cases, though, you have to be prepared to invest time and resources – you can't estimate the work based on how you've been building with Drupal 7

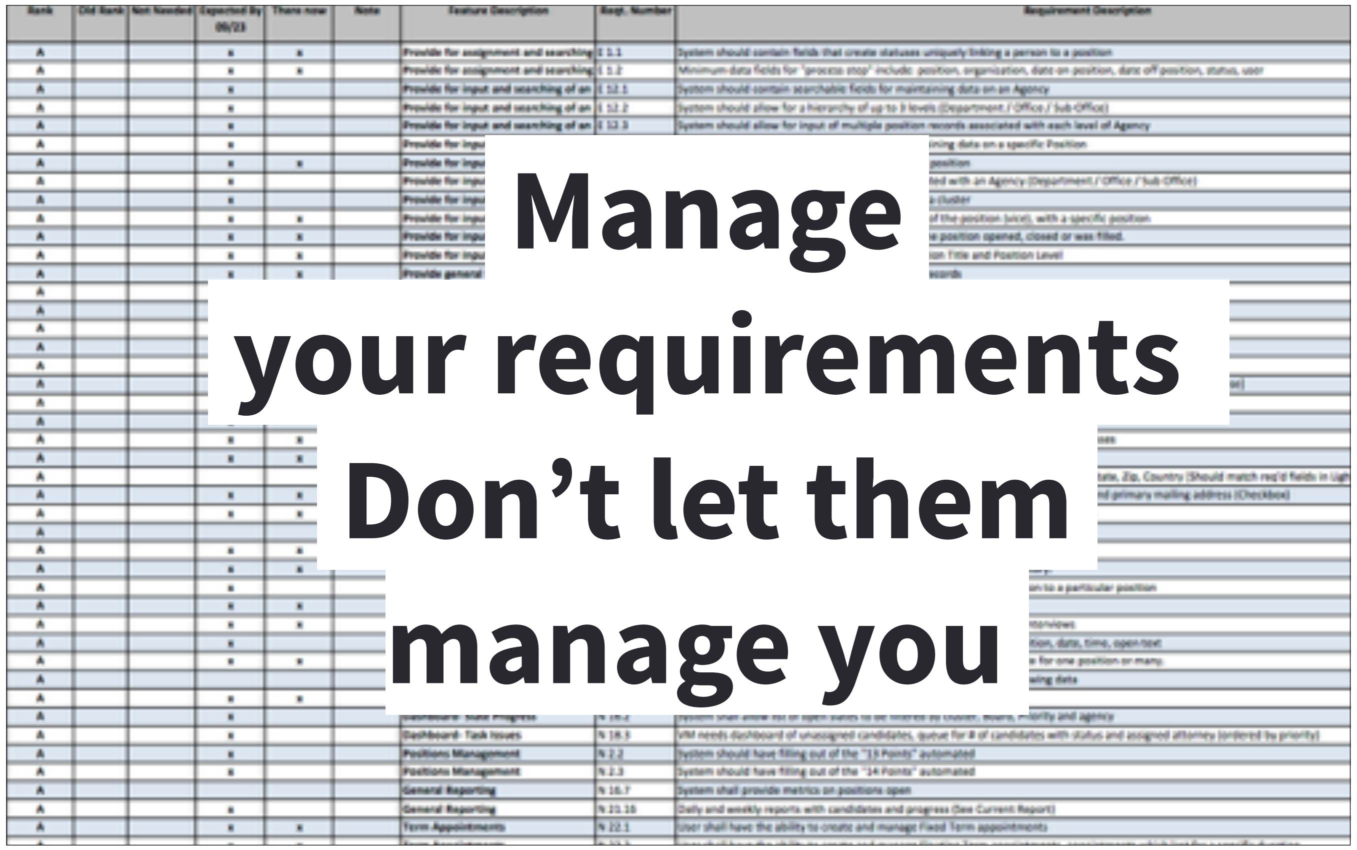


Be prepared for extra work

- Developers will need ramp up time
- Themers will need to learn Twig
- Debugging and patching for contributed modules
- Estimation of work is less certain and needs more padding

If you decide to use D8, your project plan will need to account for some extra work:

- We needed to account for the time required for developers to ramp up on the new version
- Many contributed modules did not have stable production releases when we started, and we had to spend extra time debugging and contributing patches
- We were guinea pigs – we didn’t have models and examples of how to do things so we had to prototype and iterate



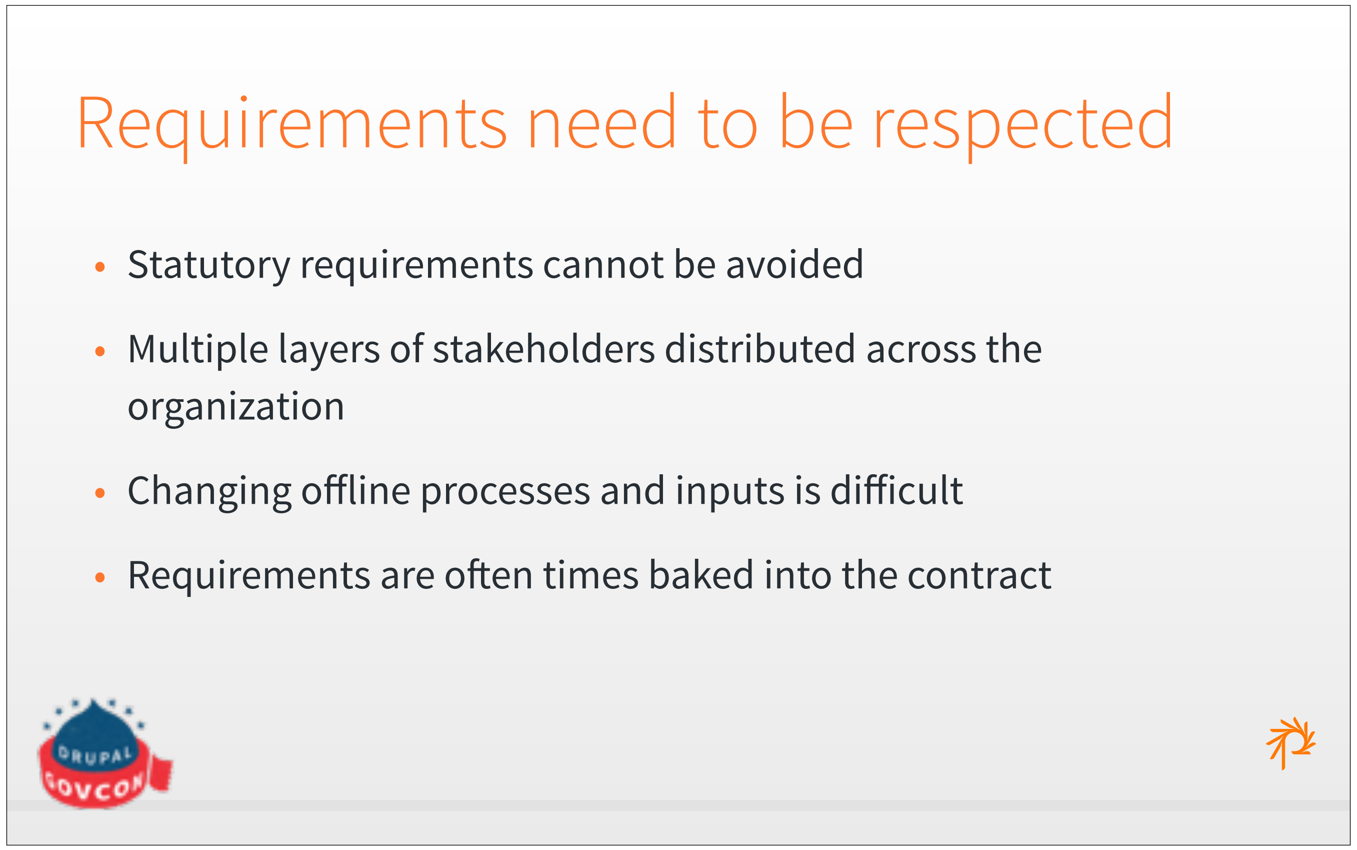
Rank	Old Rank	Not Needed	Expected By 09/23	There now	Note	Feature Description	Reqt. Number	Requirement Description
A			x	x		Provide for assignment and searching	I 1.1	System should contain fields that create statuses uniquely linking a person to a position
A			x	x		Provide for assignment and searching	I 1.2	Minimum data fields for 'process stop' include: position, organization, date on position, date off position, status, user
A			x			Provide for input and searching of an	I 12.1	System should contain searchable fields for maintaining data on an Agency
A			x			Provide for input and searching of an	I 12.2	System should allow for a hierarchy of up to 3 levels (Department / Office / Sub-Office)
A			x			Provide for input and searching of an	I 12.3	System should allow for input of multiple position records associated with each level of Agency
A			x			Provide for inpu		ining data on a specific Position
A			x	x		Provide for inpu		position
A			x			Provide for inpu		ted with an Agency (Department / Office / Sub Office)
A			x			Provide for inpu		a cluster
A			x	x		Provide for inpu		of the position (vice), with a specific position
A			x	x		Provide for inpu		e position opened, closed or was filled.
A			x	x		Provide for inpu		ion Title and Position Level
A			x	x		Provide general		ecords
A								
A								
A								
A								
A								
A								se)
A								
A								
A			x	x				ses
A			x	x				
A								tate, Zip, Country (Should match req'd fields in Ligh
A			x	x				nd primary mailing address (Checkbox)
A			x	x				
A								
A			x	x				
A			x	x				y)
A			x					on to a particular position
A			x	x				
A			x	x				terviews
A			x					tion, date, time, open text
A			x	x				e for one position or many.
A								wing data
A			x	x				
A			x			Dashboard- State Progress	N 18.2	System shall allow list of open states to be filtered by cluster, Board, Priority and agency
A			x			Dashboard- Task Issues	N 18.3	VM needs dashboard of unassigned candidates, queue for # of candidates with status and assigned attorney (ordered by priority)
A			x			Positions Management	N 2.2	System should have filling out of the "13 Points" automated
A			x			Positions Management	N 2.3	System should have filling out of the "14 Points" automated
A						General Reporting	N 16.7	System shall provide metrics on positions open
A			x			General Reporting	N 21.16	Daily and weekly reports with candidates and progress (See Current Report)
A			x	x		Term Appointments	N 22.1	User shall have the ability to create and manage Fixed Term appointments

Manage your requirements Don't let them manage you

Fairly or not, the stereotype of governmental projects is that they are waterfall in nature, come with lists of requirements, and there is little appetite for discussing or revising the requirement lists.

In our case study, we were presented with a deck of more than 600 requirements that we negotiated down to 300.

As an aside, this room after lunch, there's an excellent session on 'Successful Requirements Gathering'



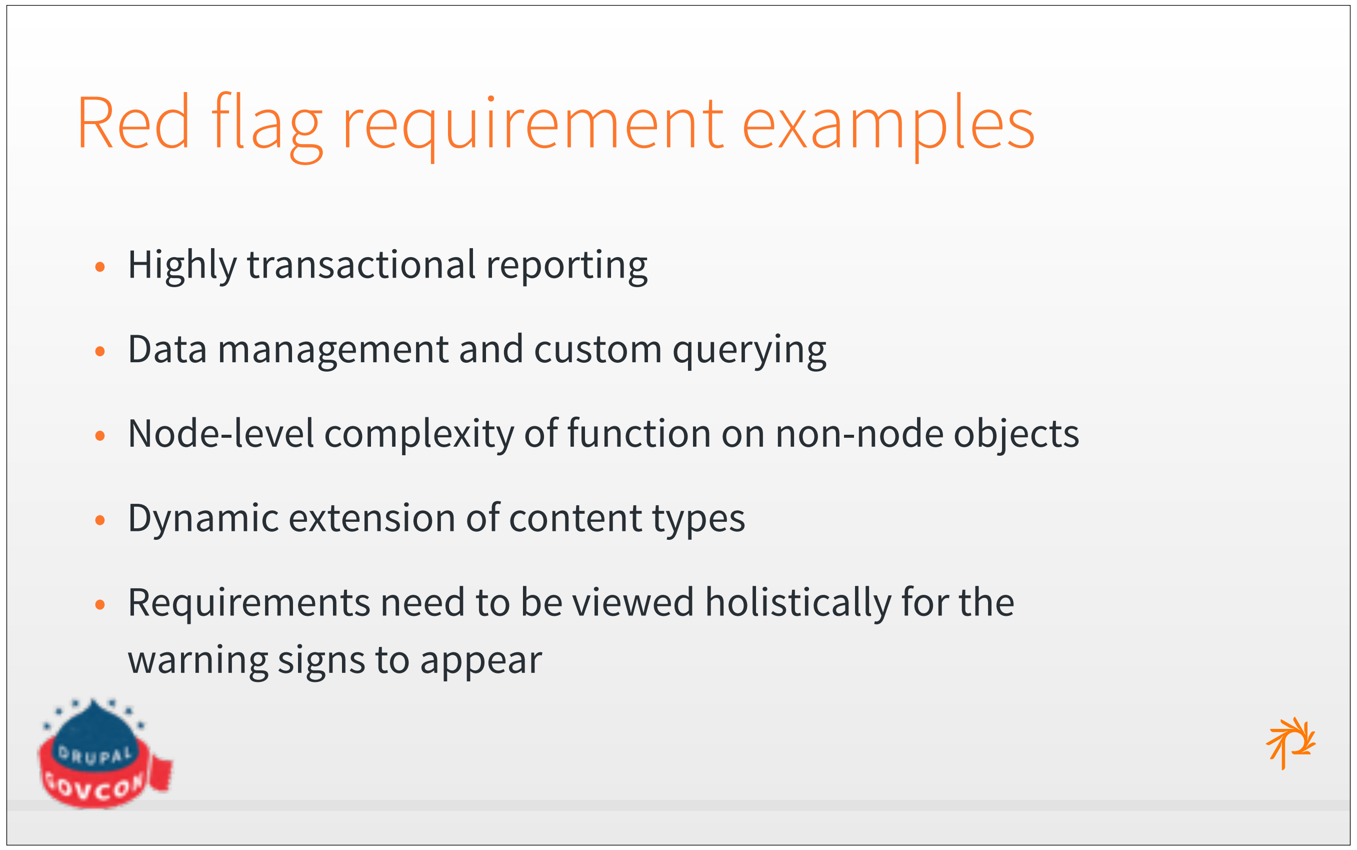
Requirements need to be respected

- Statutory requirements cannot be avoided
- Multiple layers of stakeholders distributed across the organization
- Changing offline processes and inputs is difficult
- Requirements are often times baked into the contract

From the agency point of view, we need to respect the requirements – as much we as developers love to complain about big requirements decks, they represent a body of work on the part of the client that expresses what they want and need.

There are many reasons for this, of course

- Projects are commissioned at a high level, and often times there are significant requirements that are statutorily determined.
- These are large, complex sites, and there are many layers of stakeholders, all of whom need to have input on the final product.
- There are business processes that feed into these applications that cannot be changed without serious reengineering.
- Requirements often need to be ratified internally before an RFP can be issued, and these become part of the contract, and cannot be altered without a lengthly and painful process



Red flag requirement examples

- Highly transactional reporting
- Data management and custom querying
- Node-level complexity of function on non-node objects
- Dynamic extension of content types
- Requirements need to be viewed holistically for the warning signs to appear

DRUPAL
GOVCON

From a Drupal point of view, what types of requirements should raise a red flag?

Some examples from our case study:

- Transactional reporting: The review workflows for these candidates needed to be traceable and auditable, so we needed to capture information at every step of the way, and because our system was built on entities we needed to build the reporting framework entirely from scratch. Even when you have nodes at the center of your architecture, this can be a challenge – if your requirements for reporting are that granular, there's a good chance you're generating custom logs based on actions specific to your use case
- Custom queries: There needed to be visibility into the ongoing candidate review processes, and the admins were currently able to write custom queries using their existing to-be-replaced tools – like, "show me all candidates referred from source A that are in workflows B, C, and D, and who have passed review step E but not step F". Views was not really suitable for this type of querying, and simply writing SQL queries wasn't feasible either – Drupal's underlying table structure is built to allow for extensibility, not simple query writing.

These were important requirements and there were reasons why they needed to access the data in this way. That said, just about everything we did to meet these requirements required custom code, flattened tables that duplicated data held elsewhere, and extensive user testing and iteration.

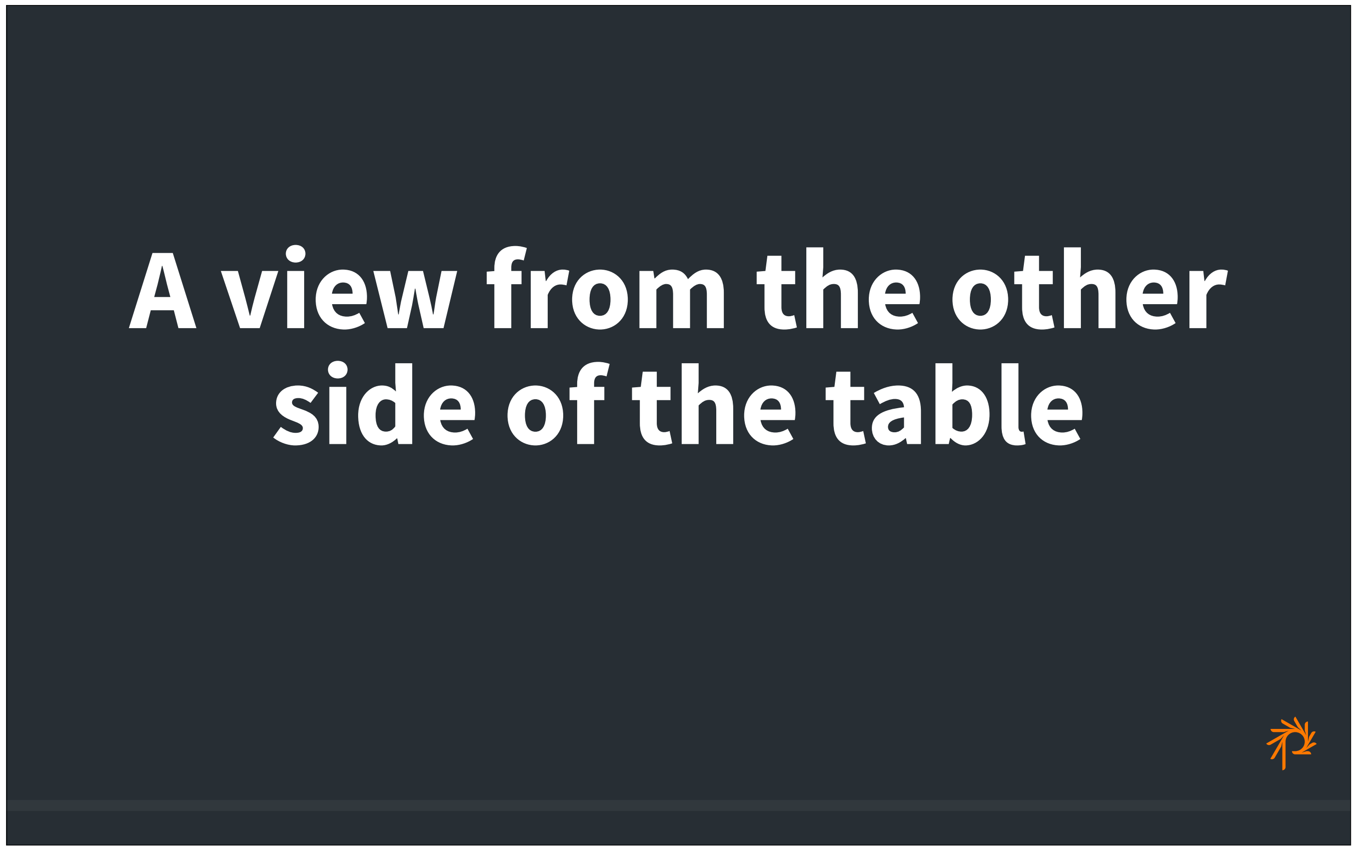
More examples:

- Dynamic extension of content types – reviewed an RFP recently that called for centrally managed content types shared across a network of collaborative sites that could be customized on the local level
- Node-level complexity of function on non-node objects –

To make this even more complex, you need to consider the implications that requirements have on each other – in isolation, you might flatten data for ease of reporting, but if there is a requirement that users be able to view a log of edits (for example), you're needing to store additional information

Assessing requirements

- Establish goals, objectives, and success criteria with the client and collaborate on priority
- Evaluate implementation options for requirements - core, contrib, custom
- Take advantage of Drupal’s flexibility and discuss alternative approaches, both online and offline

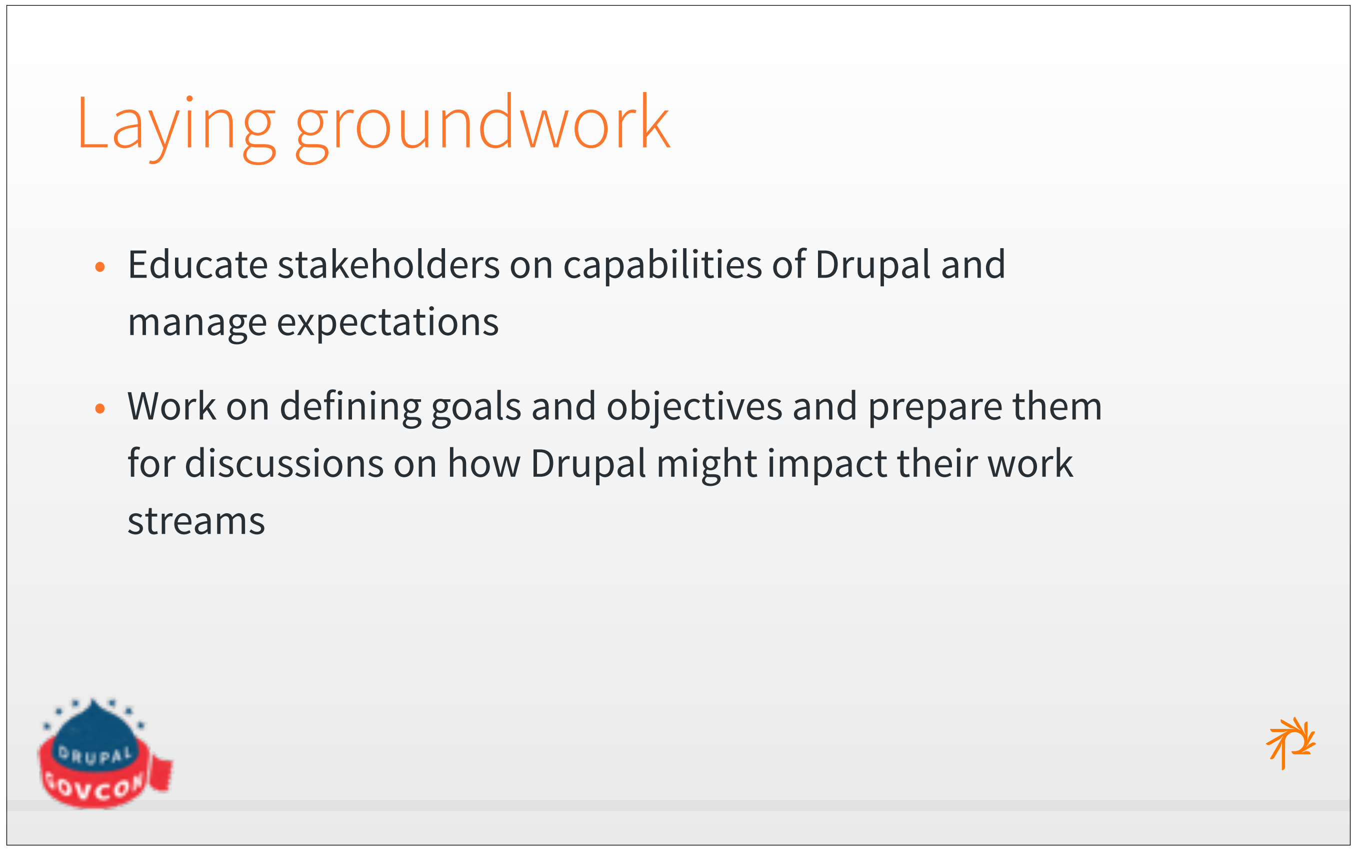


A view from the other side of the table

As a strategist, one of the roles that I have is being the advocate for the client internally. It is my job to make sure that we understand the requirements, both implicit and explicit, and if the end product doesn't meet my client's needs I've not done my job.

It's not in anyone's best interest to have a project the scope and goals of which are at odds with your chosen technical path. If you are interested in Drupal, if you have solid reasons to put it into an RFP, there are ways that you can ensure that it will be the right choice.

I think a lot of what we've talked about thus far has been pretty general and applicable to the project whether you are on the implementing or contracting side, but there are some specific things to look out for if you are looking to commission a project, if you are the client.



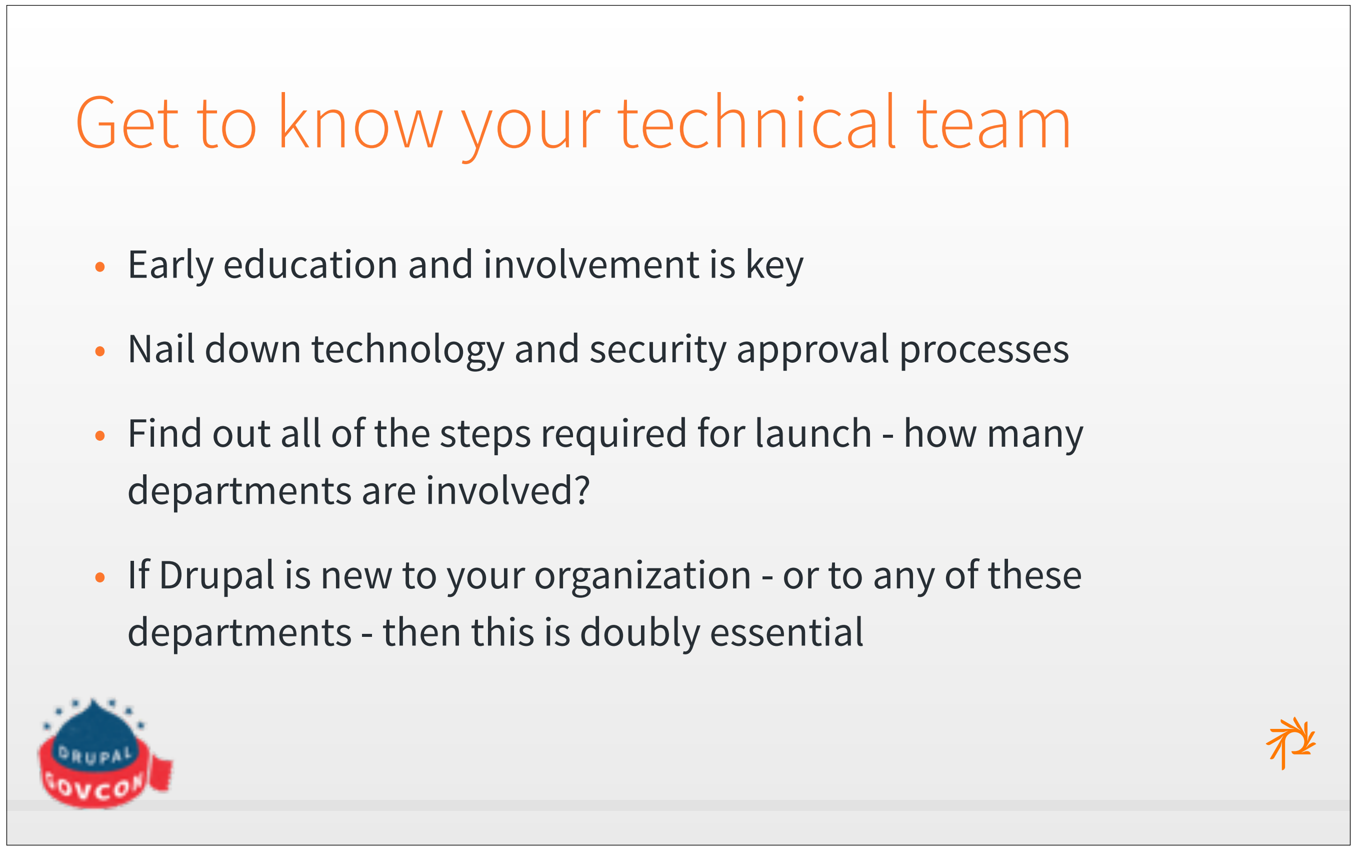
Laying groundwork

- Educate stakeholders on capabilities of Drupal and manage expectations
- Work on defining goals and objectives and prepare them for discussions on how Drupal might impact their work streams

Laying groundwork – what can you do before you

Work proactively with your stakeholders on what their goals and objectives actually are – the vendor might not always have access to these folks, so if you can, for example, help them zero in on what aspects of being able to create custom queries really are essential to their job, and which are more about comfort levels with their current tools, you can hand the Drupal vendor requirements that lead to a more Drupal-appropriate approach.

Educating them in what they can expect with a Drupal product helps as well. this makes it easier down the road when you have to negotiate with them about requirements that might not make in scope.



Get to know your technical team

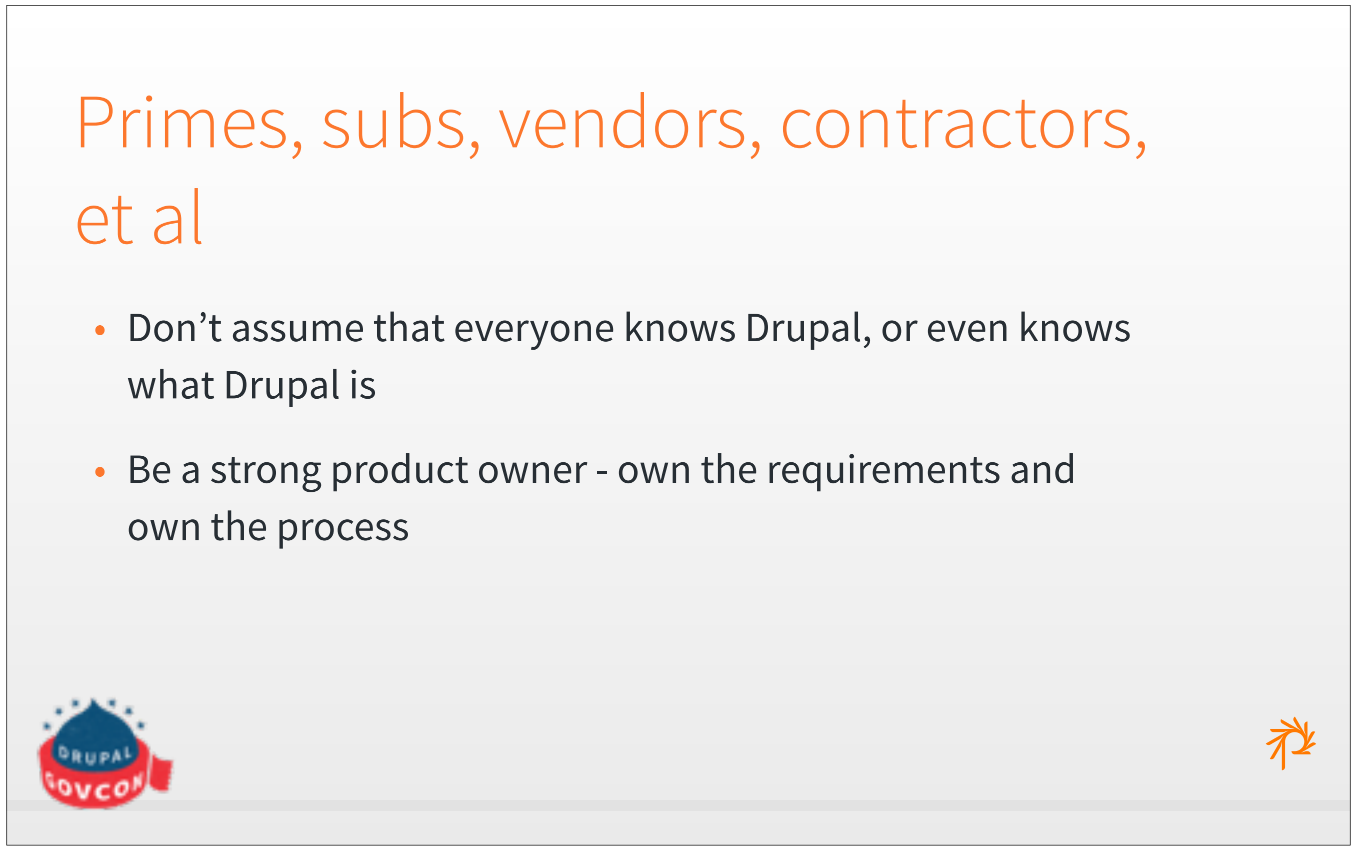
- Early education and involvement is key
- Nail down technology and security approval processes
- Find out all of the steps required for launch - how many departments are involved?
- If Drupal is new to your organization - or to any of these departments - then this is doubly essential

Get to know your technical team

There is no team more important to getting your site launched than the internal technology team at your agency, department, or organization.

- Make sure to get them involved in the project early – and if approvals are required for any aspect of the site before launch, find out what these processes involve early. They may have questions about what is needed to support Drupal, and putting them in touch with the development team early will pay dividends later.
- Same goes for the security team –
- Get to know the larger hierarchy and org chart as well, and learn what you can about their internal workings – there can be friction at certain points, so knowing what to expect is key.
- This is all best-practices – but If Drupal is new to your organization, then this becomes even more vital. For examples, you may need to bring in external security consultants versed in Drupal to validate the product. System administrators may need training.

In addition, internal IT departments at large organizations tend to be organized around the platforms that support the primary business goals – if 90% of their job is setting up Sharepoint installations on Microsoft IIS servers, then getting a LAMP stack provisioned for a Drupal site might be a challenge, even if it’s approved software.



Primes, subs, vendors, contractors, et al

- Don’t assume that everyone knows Drupal, or even knows what Drupal is
- Be a strong product owner - own the requirements and own the process

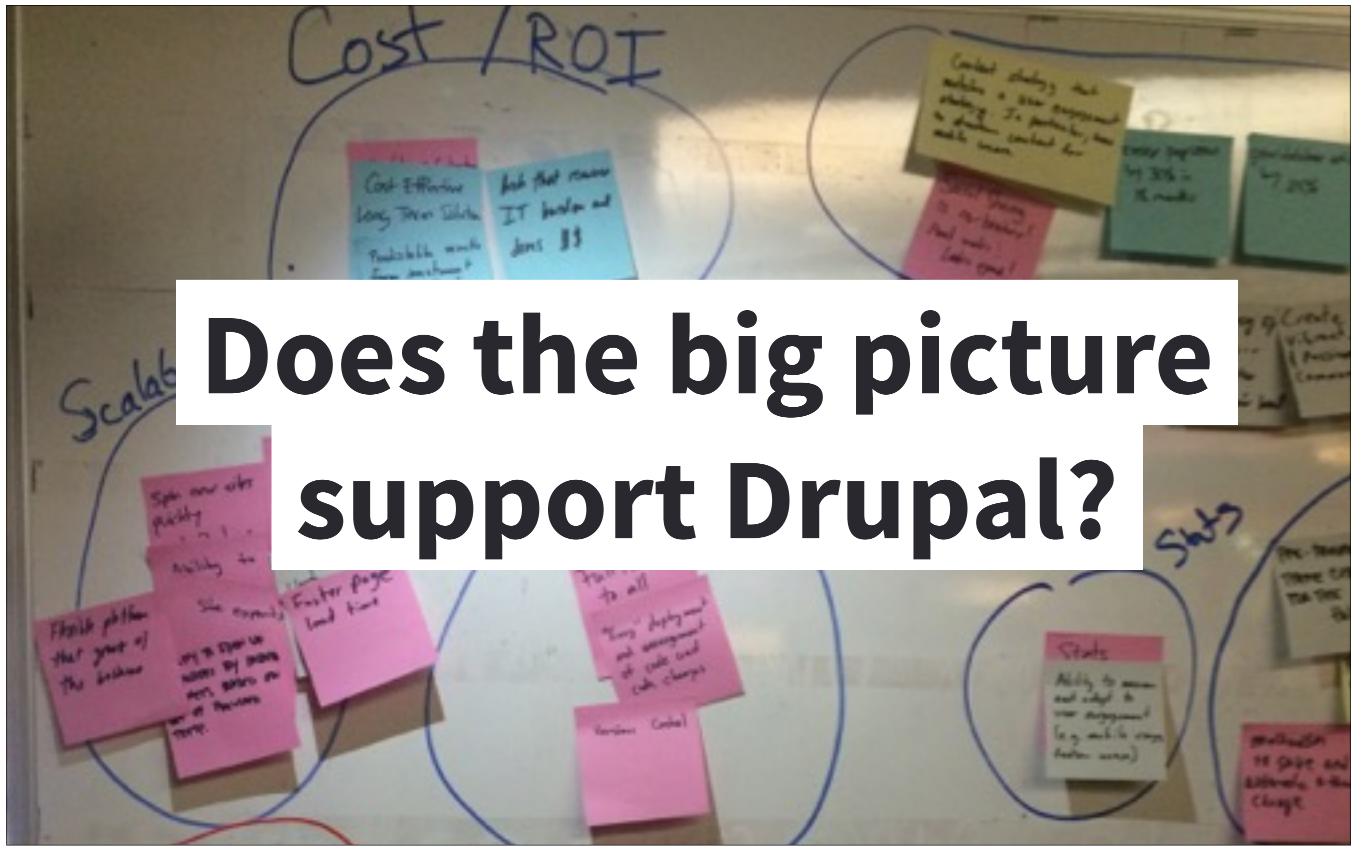
Primes, subs, vendors, contractors, et al

These are big projects, and the nature of government contracting means that there are oftentimes multiple vendors and multiple layers of contractors. We have all dealt with these challenges before, and although I have plenty to say about the larger issues I’m going to keep this focused on the salient question of how you ensure that your Drupal project isn’t affected by the structure of the vendor relationships.

Several key points:

- Don’t assume that everyone knows about Drupal – you will have a development team that is hopefully top-notch and in possession of deep knowledge about Drupal development, but you may also have separate contracts out for designers, information architects, systems administrators, and security consultants, and the degree to which they are familiar with Drupal impacts the efficiency of the inter-relationships between vendors –
    - As an example: We are currently working on a large platform build for a state government client, and working with a partner that provided design, content strategy, and IA services, while we architected and built the Drupal platform used to build out departmental and agency sites. Our counterparts at this shop are all top-notch folks, smart, really good at what they do, but they are not a Drupal shop. They made certain assumptions in their designs and IA work that didn’t match up with our architecture, and there was a learning process we all went through to adjust our respective assumptions. This relationship is a good one, an efficient one, and they have come to understand the technology intimately, but it’s not always the case.

Your role is the product owner – you can’t throw requirements over a wall and expect that the vendor team will build what you actually need without active involvement. There will be instances where you need to make decisions about aspects of your Drupal site and make sure that everyone is comfortable with it.

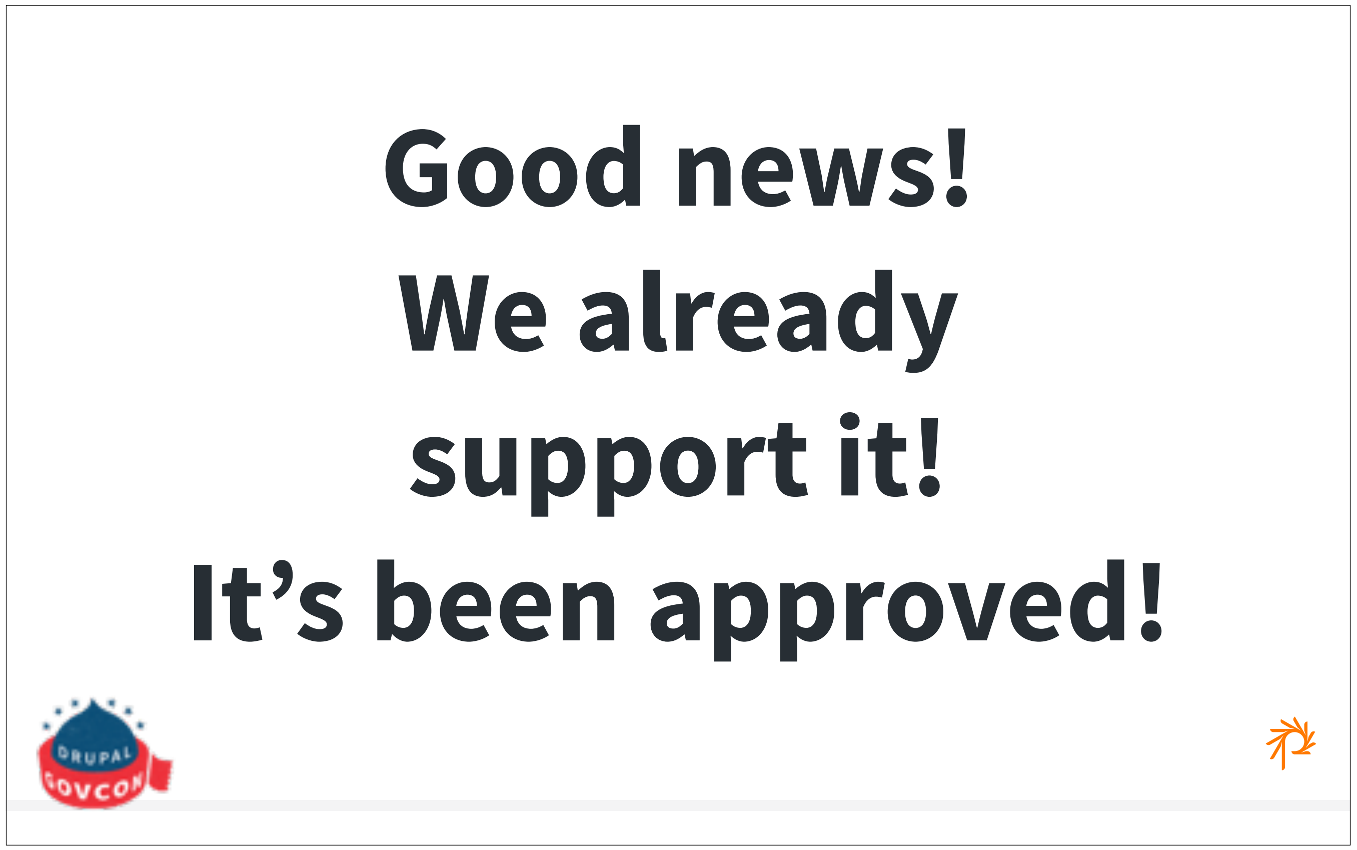


Does the big picture support Drupal?

Fairly or not, the stereotype of governmental projects is that they are waterfall in nature, come with lists of requirements, and there is little appetite for discussing or revising the requirement lists.

There are many reasons for this, of course

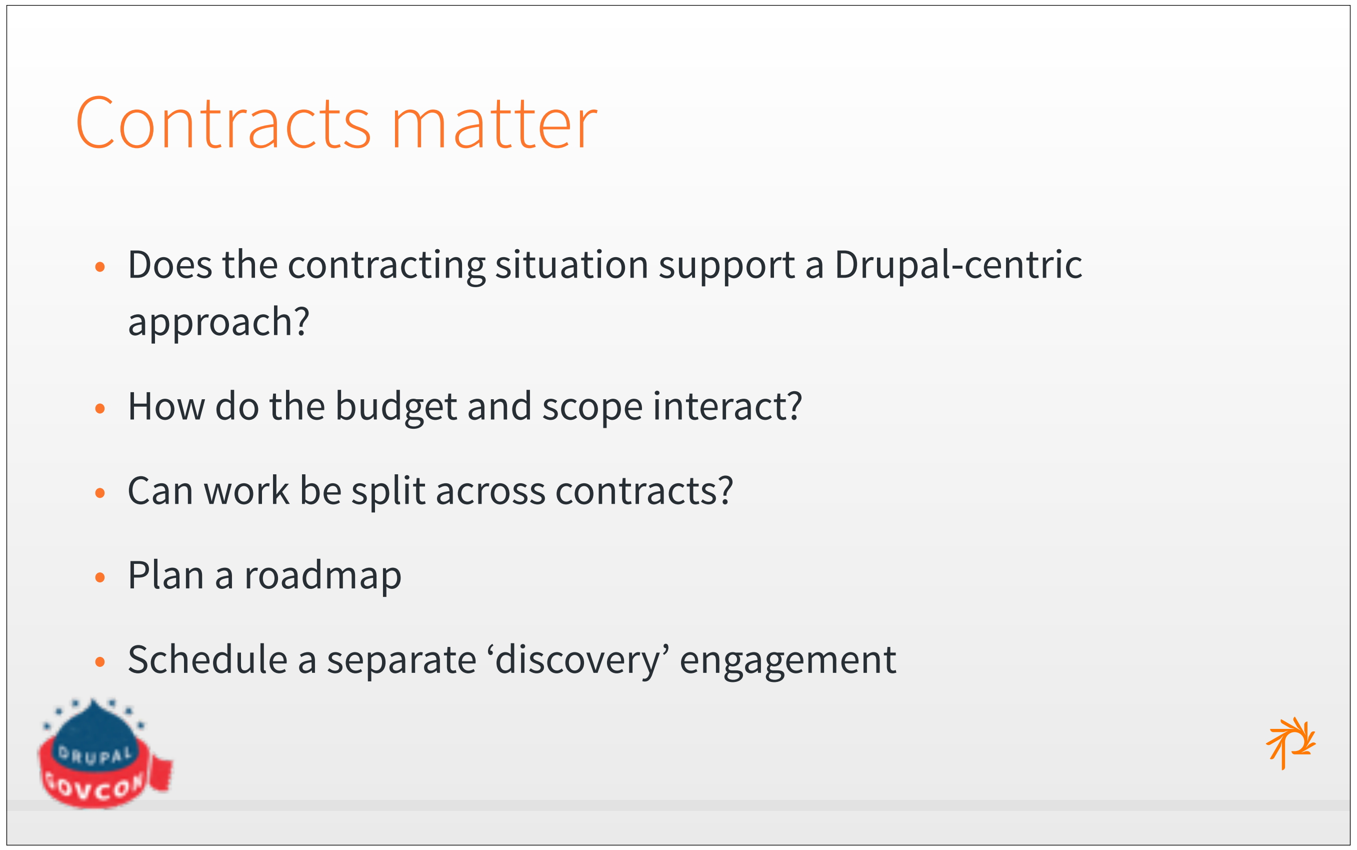
- Projects are commissioned at a high level, and often times there are significant requirements that are statutorily determined.
- These are large, complex sites, and there are many layers of stakeholders, all of whom need to have input on the final product.
- There are business processes that feed into these applications that cannot be changed without serious reengineering.
- Requirements often need to be ratified internally before an RFP can be issued, and these become part of the contract, and cannot be altered without a lengthly and painful process



Good news!
We already support it!
It’s been approved!

This is huge - Drupal is mature and has a track record, and it’s been in use for long enough that for many projects, there’s no technology approval process. Anyone who has gone through a new technology approval process knows that this can be the single biggest reason to select one technology over another.

If it’s not approved? You can work with this - someone has to be first in every organization - but you will need a strong internal champion.



Contracts matter

- Does the contracting situation support a Drupal-centric approach?
- How do the budget and scope interact?
- Can work be split across contracts?
- Plan a roadmap
- Schedule a separate ‘discovery’ engagement

Does the contracting situation support a Drupal-centric approach?

We’ve talked a bit about how, in a perfect world, the developers and the client can work together to find the best solutions to the challenges that they’re going to face. But what if there’s a prime in between that limits access to the client, and which doesn’t fully understand Drupal? This friction may mean that you are stuck with bad requirements.

How do the budget and scope interact?

If you find yourself in a situation where you’re needing to do extra work to extend Drupal to meet requirements it’s not suited for, and you’re working on a fixed scope fixed cost contract, then you’re going to take a loss. You’re not always going to have a say in how the contract is structured, but if you can focus on goals and objectives as deliverables (as opposed to requirements), you’re limiting risk.

Another way to limit risk on Drupal projects is to break risky work out into time and materials contracts if possible – this is something we try to do for migration tasks, where page count invariably inflates and content and data inevitably require cleaning up.

Plan a roadmap

Drupal is an excellent platform for iterative development – one thing I tell all of my clients when I consult with them on what they plan to do with their sites is that they shouldn’t consider launch to be the end of development. Your needs will be different in six months, in a year, in five years, and your software shouldn’t be static – it should be an evolutionary tool that grows with you. It is never too early to talk about what features you’ll need in a year, even if you don’t have budget for it yet. The flip side of course is that you should be open to moving features to later releases – this helps keep the functional profile clean and tight, and can provide a safety valve for keeping your site ‘Drupally’.

Go and get it built!

Questions?