Documentation

- Currently very confusing to find documentation
- Incentive to contribute - Core Contrib badge at least, ideally karma. Even if not possible to 80% of circle members didn't know there was new docs instead of wiki
- Possible to add tags to documentation especially by applicable by version like we have with packages
- Ideally old wiki content should not appear in Our results - archive?
- Be able to select webforms or mvc so only applicable documents are viewable
- Every page should be tagged by Umbraco version, e.g. applies to Umbraco 4.x, 6.x, 7.1 etc
- Would love for some more video content within the the documentation
- More more more code examplesF
- Automated documentation only worthwhile if code is REALLY well commented
- Would be great to have a hit counter for each page of documentation on Our so that we could see what pages are being hit but don't have good content, also could be used for search and also maybe a area showing most popular docs
- Snippets - make sure they work!
- Ability to make a comment on a page of documentation and make page as current in progress or incomplete
- Possibly create a YouTrack for Documentation project - ideally on each page of documentation on Our you could report an issue and it would create directed in YouTrack! **Awesome** idea!
- tie up documentation author and contribs, at least docs page should be able to high5ed so that ranking could be improved in search results
- Clippy for Umbraco - uClippy
- How to section - collection of different things, e.g. How do I create a image library (ideally video), how to enable x feature etc....
- If reading a forum post on Our, it would great if there was a button to flag that the content should be converted to documentation
- Distinguishing API version for samples
- Would be good to have some documentation on how to used umbraco with source control and CI
- Docs needs content to explain what is IContent, what is IPublishedContent etc - Glossery
- Umbraco tv videos would be great to be embedded in documentation and/or YouTube channel videos
- Interesting thoughts about maybe a service for user groups be able to record/live stream sessions - maybe there is app/online service that could be provided by HQ and content would feed to YouTube channel
- Twitter share button on the bottom of each docs page

Gold Partner

Some would like to see a lower cost
- for licensing benefit mostly

Some see benefit as
- free CG tickets
- sales leads from business managers

Position program as a way for partners to give back, contribute to the health of the core

Some amount of funds directed to marketing material for use in sales by partners
- whitepapers, etc...
- growing the material over time

Important to keep perspective that partners support core and make it better for all

Hard to sell program to an organization's business management - internal justification

Lead generation - hard to represent as only really potential leads

Why wouldn't you become a Gold Partner:
- money is hard to justify to business
- payment of 10k EUR hard all at once
- monthly or quarterly payment could help
- make fee billable to clients
- perhaps add a per project support payment as part of a project - passed back to umbraco

More about selling umbraco in the competitive market - sitefinity, kentico, sitecore,
...

Rocking media in Umbraco 7

- Most common complaints
  - Backup, restore, sync slow/doesn't work when image quantity grows within app.
  - Batch upload to custom media types doesn't work
  - Can't upload directly from TinyMCE
  - No automatic image resizing - users end up publishing css-scaled giant images
- Many different workflows
  - More flexible workflows for techies
  - Rigid workflows for non-techs
  - At a small scale, users want to control where their files go on the filesystem
  - At a larger scale, users want stuff to stop locking/crashing (URI immutability is needed)
  - Both hierarichal (nested folders) organization and flat, metadata-based organization are needed and used. This doesn't require the physical filesystem match.
- What's popular
  - CropUp, DAMP, ImageResizer, ImageGen
  - Some use external systems (like Flickr), or don't tell Umbraco about their media at all.
  -

Frontend workflow tool

A list of the tools we talked about at the session:
Some of the links might not be right... Sorry

- R.js (https://github.com/jrburke/r.js/)
- Mixture.io (http://mixture.io/)
- Liquid (http://wiki.shopify.com/Liquid)
- Mindescape - visual studio plugin (http://www.mindscapehq.com/products/web-workbench)
- Web essentials - visual studio plugin (http://vswebessentials.com/)
- tdcss.js (http://jakobloekke.github.io/tdcss.js/)
- grunt.js (http://gruntjs.com/)
- pow.cx (http://pow.cx/)
- Adobe Edge Inspect - aka Adobe Shadow (http://html.adobe.com/edge/inspect/)
- Browserstack.com
- Modern.ie

Umbraco for non-devs

A very lively session with a lot of ideas covering a broad range of areas.

Umbraco.com

Idea: a "solo" plan for Umbraco as a service (Umbraco.com) for users who have no idea about databases, staging/production environments, etc. Like Wordpress.com, allows users to sign up for an Umbraco instance without needing to be able to download and install it.

Code snippets and packages

Idea: a library of code snippets, rather than packages, that can be accessed from the Umbraco backend, e.g. navigation, carousels, accordion controls, etc. This could include tag-driven search, and also be voted up by the community like forum posts are. This could be wiki-driven so that community members could fix bugs they find and improve the code snippets available. Quality of snippets is important. Also handy to keep track of your own code snippets!

We could make more use of templates in XSLT and razor to allow new users to insert macros into their content without needing to be able to create those macros, e.g. wrap the current XSLT/Razor templates that come with Umbraco into macros so they are available in the richtext editor. There could be more descriptive information on what these templates do, rather than "list children" for example.

One problem with installing packages is if it breaks you are in big trouble (this is the nice way of saying it). This is the same with Wordpress and is a problem for non-technical users. It would be great to have transactional installation of packages. If a package is not compatible, it should be disabled or show a small error message (like an XSLT error at the moment) rather than break the site. This is an issue with .NET rather than Umbraco, but still needs to be solved.

Themes and starter kits

We don't want to mimic Wordpress, but the theming idea is a good one. However, we don't have a standardised markup like Wordpress does. The snippet database would be very helpful for theme developers and frontend developers. Single-action "insert into template" of snippets would be great. Usage stats could even intelligently suggest other relevant snippets based on usage of all users. A HTML preview option could allow users to see the output that they would get from their snippets so they can style it up properly or see what changes are required.

From a designer's point of view, the learning curve for Umbraco is steep, and the docs aren't as easy to follow as they are in Wordpress. It's much easier to create a theme for Wordpress by following the docs for example. Starter kits are needed in Umbraco to attract front-end developers.

The Umbraco website

Does the Umbraco website speak to end users, e.g. "Wordpress guy/gal"? Or is it too developer focused at the moment?

Help and documentation

It would be good to have inline help and tutorials for new users to talk them through the process, like Podio does. So when you first access the content section, you would get a nice popup "tour" explaining what's available to you. New users need more hand-holding - could use the dashboards for this? Wordpress does this well. The installation process could have different paths for different types of users, e.g. I'm a total novice vs. I'm an expert.
Umbraco.TV is quite technical, it would be great to have some very basic how-to guides on there for new users.

Our.umbraco.org

Our is not very visual and maybe not a place for themes and starter kits - these should be kept away from the more technical packages, and presented in a visually more appealing way to attract designers. Our is not front-ender friendly. It's very technical for some users and could be a bit friendlier for non-devs - ease them into it. Perhaps having different areas on Our to allow users to slowly enter into the more technical parts, rather than just dumping everything on them on the landing page - an absolute beginners area on Our.

Summary

According to the group, this is what is required to make Umbraco more appealing to non-technical users:

- starter kits with a library of widgets
- better docs (including tutorials for beginners on Umbraco.TV and inline tour in the Umbraco backend
- a solo plan on Umbraco.

It is a very broad problem and not easy to get to the point where "Wordpress guy/gal" easily get started with Umbraco while keeping the flexibility (which is its great strength). We don't want to mimic Wordpress but it does have some good ideas we can take on board to increase the range of users being attracted to Umbraco.

Umbraco Liquid View-Engine

Intro

With the addition of MVC in Umbraco, we can utilize the in-built view engines to allow alternative template languages to be integrated. One recent templating language being embraced by front-end developers is liquid, a language created by shopify.com to allow developers to easily and quickly create online stores. To integrate liquid into Umbraco, the open source library DotLiquid was used (http://dotliquidmarkup.org/) and converted into an MVC view engine (http://nuget.org/packages/DotLiquid.ViewEngine). A slight alteration to Umbraco was needed (though a pull request has been made + accepted so will be in the next release) to allow view engines to work correctly, but once fixed, the ViewEngine works out of the box.

To integrate, the view engine was registered during app startup as follows:

```
public void OnApplicationStarted(UmbracoApplicationBase umbracoApplication, ApplicationContext
applicationContext)
{
    var ve = new DotLiquidViewEngine();
    ve.MasterLocationFormats = ve.MasterLocationFormats.Concat(new[]
    {
        "~/Views/_{0}.liquid"
    }).ToArray();
    ve.ViewLocationFormats = ve.ViewLocationFormats.Concat(new[]
    {
        "~/Views/{0}.liquid"
    }).ToArray();
    ve.PartialViewLocationFormats = ve.PartialViewLocationFormats.Concat(new[]
    {
        "~/Views/Partials/{0}.liquid"
    }).ToArray();

    ViewEngines.Engines.Add(ve);

    DotLiquid.Template.RegisterSafeType(typeof(RenderModel), new[]
    {
        "Content"
    });

    DotLiquid.Template.RegisterSafeType(typeof(IPublishedContent), new[]
    {
        "Id", "Name", "Url", "Children"
    });
}
```

Additional view locations were registered to allow it to work with the Umbraco standard, and the umbraco models were registered. Once problem to think about is that liquid is a strict language, so only specified properties can be accessed, so some thought needs to go into what should be exposed.

Discussions

After the demo, attendees were asked their thoughts on the use of Liquid (/ potential other view engine languages) and whether they could increase adoption.

Members agreed the option of a liquid view engine would be very well received by the front end community due to it's easier syntax over razor, especially when used in combination with external tools such as mixture.io (as mentioned in "meet me in the browser" session) which allows developers to work with liquid during front end development, and would allow these to be just dropped in to an Umbraco solutions rather than requiring a conversion step.

To make alternative view engines known about (ie, we don't want front end devs to try Umbraco, but give up because they don't like razor, but didn't know the view engine could be replaced), the umbraco installer could be updated to allow developers to set their primary view engine.

Updates would also need to be made to the umbraco template editor to take into account the view engine used, plus allow alternative buttons / syntax highlighting to be incorporated based upon the type of view the user is editing.

By using the liquid view engine, it also allows larger teams to create paralel workflows allowing a data "contract" to be defined in JSON at project startup, then front end devs can build the ui against it, and back end devs can setup umbraco to create the same data contract.

In this situation, the prefered structure of a project would be to use hijack controllers in Umbraco to hijack routes, and then use view models to just pass down the required information for the page.

For those unfamiliar with liquid, it is suggested to checkout the docs on github https://github.com/Shopify/liquid/wiki/_pages aswell as setup a trial on Shopify.com to see it's use in real life.

Packages and the developer experience

Majority of the participants would like to see implemented as Nuget packages

- Files are easily implemented
- But what about post install actions?
- -> Maybe - Signalr communicating with Umbraco service

Nuget pack could also run without Visual Studio

What does the core team feel?

- Core want's to stick with current format.
- Not easy with current solution

What about standards

- Wait for extensions point?
- But they need to be there - to tell new package developers
- Review existing packages, to see what's done?

Backend registration to enable unregistering of packages

Upgrade paths and notifications is also needed

Current state prevents people doing development through VS from using packages

- Because lack of trust in being able to upgrade Umbraco later
- -> better to offer smaller set of possibilities and and enforce some standards, to increase trust level

Drupal model

- Dashboard in backend
- Possible to turn packages on and off
- Notifications when packages are upgraded

Extensions points

- The more extension point the better
- Look to what is possible with Wordpress hooks and filters
- IPackage interface, to make it possible to detect and react, during application start

Isolation !

- There might really complicated issues around uninstalling
- Maybe a simple standard, that most packages can live with,
  and then have a flag telling that this package doesn't adhere to standards,

Code First, Code Generation and Strongly Typed Queries

Discussing whether to implement a bunch of stuff to support Code First, Code Generation and Strongly Typed Queries in the Core versus have extension points that packages like uSiteBuilder could utilize to provide Code First.

Shannon suggests that “only” strongly typed queries is natively supported in the Core, but have kind of a description/standard/conventions for models, so that the Core can use those (seamlessly).

The Core team has made proof of concepts for Code First, Code Generated models and strongly typed queries, so we know its definitely doable.

The Core team agrees that we should enable both Code First and Code Gen, but the model that is created should be the same regardless of which approach the developer preferes to use.

Stephane explains the Proof of concept he has done using IPublishedContent, which inherits from a base class and take IPublishedContent through the constructor to map a collection of properties to the properties on the custom strongly typed object.

Pete Duncanson explains how uSiteBuilder works with exporting classes from doc types in Umbraco.

If the Core team can better support the changes to improve something like uSiteBuilder it will be easier for the uSiteBuilder team to continue the development of it and make it better/faster/stronger. Everyone is encouraged to help out with the uSiteBuilder codebase.

Shannon suggests that the Models and how they are defined and decorated with attributes is defined in the Core, and provides hooks to stuff like uSiteBuilder to provide code first capabilities.
Code generation should be supported in the Core to actually enable strongly typed query, and then Code First could be provided externally to enable Code First instead of Code Gen. for creating the models that are needed for the query part of things.

Using a build provider is suggested, as it worked very well in another CMS (Kentico). Idea is to build models on the fly for the code generation part.

Shannon had an awesome idea: register attributes on PropertyValueConverters that defines the type that it should return.

We went one minute over time :-/

Package Standards

Notes coming. Just need to transcribe this:
https://dl.dropboxusercontent.com/u/901850/CG13%20OpenSpace/Package%20Standards.m4a

Visual Studio Umbraco development

Notes coming. Just need to transcribe this:
https://dl.dropboxusercontent.com/u/901850/CG13%20OpenSpace/UmbracoStudio.m4a

Umbraco In North America

Build Community

Leverage Microsoft evangelists (hanselman or scottgu)

Twitter hashtag (#?)

LinkedIn Group? Who manages?

Umbraco newsletter - hq with with marketing

Nauug.com as the central place?
- location aware - suggest a user group

Divide and conquer - start small and build

Virtual meet ups monthly - hang outs

Microsoft can help with a location

Festival

- charge a nominal fee
- hands on day?

Having a festival in an attractive location could result in better attendance with getting European Umbracians to come.

SF, NYC, Chicago, Orlando, Las Vegas*

Use the existing Google Group to organize
- get with Paul to coordinate with the Umbraco hq events

Kickstarter type funding

Optimization session

Umbraco takes a long time to start up - slow response time criticism from clients-
Phil Harvey - output caching as static html & re-set after re-publishing. Not useful for macros. - hijack http pipeline. Hooking into the umbraco events.
Marc - Application pool techniques
Caching macros - static html cache.
Morten - performance optimisation: in back end only load up what is needed.
Anders - is performance a focus at the HQ?
Morten - optimisations closely related to umbraco as a service
Stephan - XML Cache replaced by object cache, navigation of XML document usually for menu building etc.

Backword compatibility… Optional.

Cache replacement from version 8. Marl properties as 'unsearchable' (e.g. in DOcument types)

Graphical performance logging tool / package. Mini profiler will be available from 6.1

Database optimisation - peta poco.
Put
dex on version cmsPropertyData, some indexes on the table
Package to create indexes or have it in core. But do not go crazy and index everything.
Optimisation: VS long running queries tool, management view for unused indexes.

Could backwards compatibility be a contributing issue.

Memory footprint issues - in the pipeline, memory hit during preview - and for each user as well (e.g. 5 users)

Should there be a communtity user group to look at issues to feedback to the core team.

Package module to monitor performance and send data back to the Core.

Fragmented server: e.g. separate front and back office deployments. Could be achieved with load balancing (but may need to turn off macro caching)

Load balancing: sticky sessions + single back office server.

Where is load balancing stuff going? Shannon has done a lot of work and load balancing is improved in version 6.1

Cf. Drupal: connect to many events in the lifecycle - e.g. re-routing many events, adding timers etc.

Improving the database & its performance? Review Indexes & query. Also are queries are asking for too much information.
Also DB slowed down by too many versions. Also hits versed properties.
Adding stored procedures.
Trashcan deleted = slow b/c also need to check for files.
Background task for deleting unused media files.

Optimise how content is saved in the DB.

Application start up - in version 6 there is a certain order and easier to find bottlenecks.

Sharing DB performance output.

Getting Started with Umbraco

Summary

- **We can still make improvements to make it easier to get started**
- **Would be great to have ONE place to go for getting started (maybe wiki page is good enough)**
- **We could make better use of Community blog posts / screenrs etc. (maybe linked from Our Umbraco)**
- **Installer could be improved with proper error messages**
- **Umbraco Version numbers attributed to Our Topics would be helpful**
- **Umbraco TV is good when up to date and has new content.**
- **Good 'official/ recommended' Community starter kit would be welcomed.**
- **Packages are getting out of date on Our, makes it harder to get if your first package install fails.**
- **Would be good to have an aim of getting people started within X minutes**, **this**

could then be measurable

Actions taken on board

- **Will look at what is available on Our for getting started, look at setting up Wiki page if not.**

- **Will look at Starter Kit / Theme options, including Standard Dataset**

Open Space Notes

Generally it's still felt that getting started with Umbraco is harder than it needs to be.

The feedback given was that most people google around trying to work out how to get started, there are great resources out there but often they are out of date and also being on external (i.e. not our.umbraco.org or Umbraco.com) new starters cannot be sure on the quality. Information is out there, where is it? Is it valid, good, in date, right version?

It was felt that it would be great to have ONE resource for getting started, a Wiki page on Our could be good enough, however this should include relevant posts / tutorials etc. from the community and external pages rather than just Umbraco HQ content.

Umbraco TV (or at least the new Beta) was said to be useful but sometimes out of date.

There needs to be Version numbers our.Umbraco.org (again) attributed to Forum posts, blog posts etc. Users very frustrated to try something which doesn't work in the their Umbraco version.

It was felt it is totally reasonable to expect a new user to install Umbraco and have a one page site up and running within an hour, if the install fails, or people struggle it's assumed they would leave.

Installer

The installer needs to be as bullet proof as possible (for example, check DB permissions then inform on fail, rather than the infamous fail at 5%).

It was felt that most people would be happy to start with a good starter kit, rather than documentation, in the past CWS was useful, Runway not so much. With the current 'starter kits' it was felt unsure if they work, with what versions, the most popular one has a bug in the slider (since forever).

Standard Skinning Dataset Simple - One reference point

It was suggested that there should be a 'standard' 'official' starter kit, that users can use as best practice, we know there is many in the community, but would be good to have a standard one so people can reference it when they have problems (I'm trying to get the first 5 FAQ's in the starter kit, i've tried this, it doesn't work etc.) This would need a 'standard and agreed' doctype / properties etc..

Codegarden

Was felt that it missed info for new users, especially the first day, was too heavy, people who had been to codegarden pointed to Doug's "Getting started with Umbraco'

Packages

Frustration with packages, loads of out date, 'works with' umbraco x.x.x' is rarely filled out (not sure what solution is here).

Feedback

Would be great to get feedback on these open space notes rich.green@umbrellainc.co.uk
twitter: richie_green
or just use #Umbraco hash

Codegarden 2014

In this session we reviewed how Codegarden 2013 was, and how we could improve it for next year. There were 12 attendees, a mix of people that were for the first time to CodeGarden, and some that has been here for quite some time.

First of all: Everybody loves CodeGarden…………., but there could be some improvements :)

Idea: Start with open session, instead of ending with it. First-timers have a much easier way to start conversations and the first-timers loved the open session and finally got into discussion with other community-members.

Remark: Some presentations weren't really "Umbraco", but more focused on general development. Most of the people thought that the Umbraco-festival should always have an Umbraco-facing presentation. It shouldn't be hard to do that probably, if you tweaked some of the presentations a bit.

Remark/Idea: Classify a session to what audience it's targeted: Tech/Front-End/Novice Umbraco-user/Experienced Umbraco-user/Hardcode coding/Showcase/Workshop/etcetera. Now that's not clear for the audience and the speaker and a lot of presentations were high-level presentations instead of going into the deep.

Idea: More trackwise presentations

Remark: The group at this discussion is not really a representative group for all of CodeGarden. Maybe a enquiry after CodeGarden?

Idea: Give the speakers some guidelines for their presentations.

Idea: Have some Pecha Kucha-presentations (20 slides, 20 seconds each) or micro-sessions. In these presentations you can see multiple speakers in a fast pace + Everybody has got something to tell what they did with Umbraco last year that is interesting for other people.

Idea: Umbraco Speeddating. One on one for a few minutes and go on to the next person. Remark of somebody who has been around for a while: The format of CodeGarden hasn't changed, but the group of attendees has changed (and got mixed a lot).

Idea: It should be interesting to have more sessions about where Umbraco is going (not only one session on AngularJS and one on Umbraco as a Service).

The communication should probably be improved; now most of the schedule-changes/information is broadcasted via Twitter, but the WiFi wasn´t that good, so a lot of people didn´t get those updates. Just go back to basic; use Post-It's and Whiteboards J. Or show a constant #CG13-twitter stream somewhere.

Idea: Get a FAQ-wall where you can put on your questions on post-it's. When a part of the wall is full, let them be answered by an Umbraco-pro (or a couple of Umbraco-pro's) in a session.

SEO & PPC with Umbraco

- Should urlrewriting be included in Umbraco? Is it really needed?
- Default canonicalisation rules would be good to have in the web.config
- Is it possible to use request pipeline extension with handlers?
- The default URL Alias option in Umbraco is bad in terms of SEO and duplicate content - since the same page can have multiple URLs. Is this ever really needed or used? | Should it be removed to help users avoid duplicate content issues?
- When will the ShortURL package be released for vanity URLs? (project by attackmonkey/epiphany)
- What A-B and multivariate testing tools can be successfully used with Umbraco? Google Experiments & Visual Website Optimiser recommended.
- What are the best practices for a set of page properties for SEO on an Umbraco page? We already have some default 'Properties', but since SEO is key to almost any site, should there by a packaged collection or starter kit?
- A range of SERP (Search Engine Result Pages) result preview datatypes would be great, Google, Yahoo, Bing, Yandex, etc - will be easier to implement with Belle and Angular.
- A package to track inbound search terms, direct visits and PPC visits would be useful - especially if this information was accessible in forms and code - so you could create logic based on the keyword a visitor lands on your website with. Epiphany currnetly building & testing a package that offers this feature. Custom querystring variable tracking would also be great, e.g. "?x=".