

# OpenSource@UW-Madison

in 15 minutes

Jim Helwig  
MyUW Lead Strategist

Working for UW on the MyUW team for 15 years, currently as the lead strategist. This talk is about getting involved with the development of open source

Questions?  
Examples?  
Add them now!

<https://goo.gl/AzwmES>

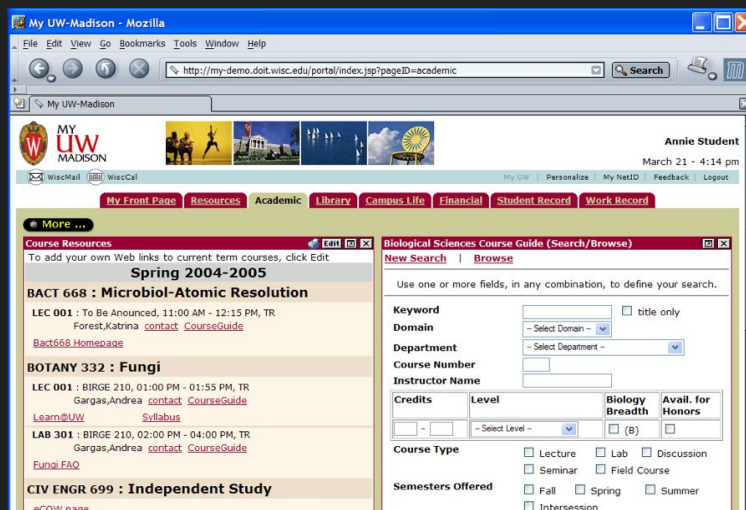
Questions? Examples? <https://goo.gl/AzwmES>

While we won't have time for Q&A today, I do want to provide a little interactivity by having a google doc available for you to ask questions or even more importantly, provide examples of open source projects you are working on. This document will continue to be available after the presentation and we'll work on answers in it.

# MyUW as example

Throughout the talk I'll be using MyUW as an example of how deep into open source you can go.

# MyUW circa 2002



Questions? Examples? <https://goo.gl/AzwmES>

When we first launched MyUW over 15 years ago, we used the leading commercial product at the time. We had quite a bit of success, although this screenshot with the turn of the century look and Mozilla browser sure looks dated.

## The vendor squeeze



Questions? Examples? <https://goo.gl/AzwmES>

Even though we were successful, we were feeling squeezed by the vendor through forced upgrades, the requirements of additional equipment&licensing, poor support, lack of input to features, and a lack of affinity with higher ed

## Evaluated options



Questions? Examples? <https://goo.gl/AzwmES>

So we evaluated several options including sticking with the vendor, switching to another commercial portal platform, or leveraging the new open source platform, uPortal, designed by higher ed for higher ed. In the end we choose to switch to uPortal, primarily for the greater control and the affinity with higher ed.

## All in...

Enhanced product

Conference presentations

Contributed enhancements

Dev meeting hosting

Supported staff involvement

List hosting

Commit access

Steering committee

Project tech lead

Apereo Foundation board

Questions? Examples? <https://goo.gl/AzwmES>

And when we switched to open source, we went all in. We immediately enhanced and extended the product. We then contributed all the enhancements back to the core project. Our management supported our being involved with the project and our staff had commit access. In fact, we hired the project's technical lead with the understanding that approximately 25% of his time would be spent on general project work.

uPortal falls under the Apereo Foundation and we have presented at their conferences. We have hosted developer meetups and even hosted the project's email discussion lists for many years until they were migrated to google. For many years we have had a representative on the uPortal steering committee and I have served on the foundation board of directors.

# MyUW today

The screenshot displays the MyUW dashboard interface. At the top, there is a red navigation bar with the MyUW logo, a search bar labeled "Search MyUW", and a user profile icon for "Jim". Below the navigation bar, the dashboard is organized into a grid of widgets. The top row includes "Payroll Information" (listing earnings statements for 10/27/2016, 10/13/2016, and 09/29/2016), "Weather" (for Madison, Wisconsin, showing a 3-day forecast), "Email" (with an envelope icon), and "Course Services" (with icons for Learn@UW, Classlist, Classroom Scheduling, and Grading Policy). The bottom row includes "Personal Information" (with a person icon), "Benefit Information" (highlighting "Annual Benefits Enrollment" with "16 days left to enroll"), "HR, Payroll and Benefits News" (listing insurance and enrollment news), and "Course Guide" (with a search bar and options to search by term, instructor, subject, or degree plan). Each widget has a "Launch full app" or "See all" button at the bottom.

Questions? Examples? <https://goo.gl/AzwmES>

For over a decade we have been heavily involved with the uPortal project. We have also created new projects and are moving them under the Apereo umbrella. While we have make use of many open source libraries and in some cases we have submitted patches. All of this has helped enable us to make MyUW what it is today.



# Benefits

That gives you a bit of a flavor of how MyUW is involved with open source. I want to speak now to the benefits of this.

# Benefits for your unit

I'll first speak to the benefits that may be realized by your team or unit.

# Easier to share code

Questions? Examples? <https://goo.gl/AzwmES>

Making your work open source makes it that much easier to share libraries or services with different projects throughout your unit. It's much easier to be aware of work that others are doing and to take a look and see if it is relevant to your work.

# Easier for others to contribute, extend, fix

Questions? Examples? <https://goo.gl/AzwmES>

Once others are using the code, it is much easier for them to contribute to the ongoing enhancement, maintenance, and support of the work.

# Encourages collaboration

Questions? Examples? <https://goo.gl/AzwmES>

This helps encourage collaboration in general. Even if everyone is still working on their own projects, they are more likely to see areas where they can work together.

# Encourages better code

Questions? Examples? <https://goo.gl/AzwmES>

Another benefit of this collaboration is that you have the sense that others are looking at your work and that helps encourage you to do the right thing. That may mean better organization, refactoring, or documentation. You can benefit by seeing the work that others are doing as well.

# Encourages mindset of delivering value

Questions? Examples? <https://goo.gl/AzwmES>

This sense that others are using your code also helps drive a mindset of delivering value. That in turn increases pride and job satisfaction.

# Encourages openness to feedback and new ideas

Questions? Examples? <https://goo.gl/AzwmES>

In order for this collaboration to be successful, it is important to be open to feedback from others. Beyond code reviews, others can often be a great source of ideas for new features or refinements.



# Benefits to individuals

There are also benefits to individuals themselves.

# Easier to start contributing

Questions? Examples? <https://goo.gl/AzwmES>

Open source projects are often easier to get started with. You can look at the code. If you want to get some hands on experience, you can find a small bug or issue and submit a pull request.

# Public portfolio

Questions? Examples? <https://goo.gl/AzwmES>

Once you start contributing to a project, you are building up a public portfolio. For student employees, this can be a differentiator when they start interviewing for their first job. For more experienced employees, it can provide a real sense of their ability to code - and to get along with others on a collaborative project.

# Clarity on individual contributions

Questions? Examples? <https://goo.gl/AzwmES>

The nature of open repositories makes it very clear who contributed what.

## Benefits to the university

Finally, there are some additional benefits to the university or greater community.

# Aligned with the Wisconsin Idea

Questions? Examples? <https://goo.gl/AzwmES>

First off, it aligns very well with the Wisconsin Idea of making a difference beyond the campus borders.

# Publicize team's work

Questions? Examples? <https://goo.gl/AzwmES>

It can also help publicize the good work your team is doing. This can be very valuable when recruiting new employees. Applicants can get a good sense of the type of work your team does.

# Increase credibility, influence

Questions? Examples? <https://goo.gl/AzwmES>

This public visibility helps increase the credibility of the university as a leader. Involvement with open source can help influence progress and future development in particular areas.



# Demonstrates an openness to openness

Questions? Examples? <https://goo.gl/AzwmES>

Open sourcing your work is another example of how the UW is committed to openness.

# Expand use beyond campus

Questions? Examples? <https://goo.gl/AzwmES>

Making your work open source makes it much easier for others to actually start using the software in their own environment.

# Easier to build community

Questions? Examples? <https://goo.gl/AzwmES>

As others start to use the work, it is easier to build a community around them to further enhance the project.

# More avenues for support

Questions? Examples? <https://goo.gl/AzwmES>

And this larger community increase the ways the software can be supported. For uPortal, we can ping the user list to see if others are seeing a similar issue and how they addressed it. There is also the opportunity for commercial providers to offer paid support.

# Challenges

So those are a lot of benefits of making your work open source, but let's look at some of the challenges.

# Leadership understanding and support

Questions? Examples? <https://goo.gl/AzwmES>

While we have strong support for our engagement with open source and a fair level of understanding from our leadership, this may not be the case in a unit that is first venturing into this. While some open source work could be done in a skunkworks manner, it is best to have some frank conversations with your leadership as to the benefits of going open and enlisting their support.

# Developer understanding and support

Questions? Examples? <https://goo.gl/AzwmES>

It is also important to ensure your team understands the value of pursuing open source. There is a certain amount of overhead and it could be tempting to revert back to a non-collaborative approach. If an employee comes from the private sector, the whole concept of being open may be foreign to them.

# Staying the course

Questions? Examples? <https://goo.gl/AzwmES>

It is important to maintain your open practices over time. Keep contributing back. Keep documenting. Keep engaged with the growing community.



# Balancing local demands and greater good

Questions? Examples? <https://goo.gl/AzwmES>

It may be challenging at times to keep the right balance between local priorities and community priorities. The UW is paying our salaries and this typically drives the work, but we try to keep aligned with emerging needs of the greater community as well.

# Tips for success

So if you are interested in trying this, here are some tips from our decade and a half of experience.

# Start with open

Questions? Examples? <https://goo.gl/AzwmES>

Open your project as soon as possible. Ideally from the beginning. Too often we see people saying they intend on opening their project as soon as they have their first release stable - or as soon as they get it documented. There can be a good bit of work going through an existing code base and refactoring it so others can use it - if you are open from the start you can avoid that. Also, the more you delay being open, the more potential feedback you miss out on.

# Choose an appropriate license

Questions? Examples? <https://goo.gl/AzwmES>

Make sure your code is appropriately licensed. GPL and Apache are common but you need to determine how your work might be packaged and deployed with other work. Go to <https://choosealicense.com/licenses/> for more information.

# Don't fork

Questions? Examples? <https://goo.gl/AzwmES>

Avoid forking. Try to use the official releases. If you fix a bug or make a modification, contribute it to the main project and cut a new release. Merging new releases into a local fork is a lot of work.

# Use open standards

Questions? Examples? <https://goo.gl/AzwmES>

When possible, use open standards. It makes it easier for others to use your work and increases the likelihood of compatibility with other projects.

# Use other open source libraries

Questions? Examples? <https://goo.gl/AzwmES>

Where possible, use other open source libraries. Concentrate your work on what is a differentiator, use other projects for more commodity capability. If it doesn't quite meet your needs, submit a pull request to that project.

# Choose common tools

Questions? Examples? <https://goo.gl/AzwmES>

Use common tools and infrastructure. This increases exposure and the likelihood that others will be able to easily contribute. uPortal uses github and google apps.



# Find the right community

Questions? Examples? <https://goo.gl/AzwmES>

You are more likely to grow your project if you can plug into an existing community. There are a number of foundations that foster open source projects. They can help with the overhead of a project. uPortal is part of the Apereo Foundation.

# Be publically active

Questions? Examples? <https://goo.gl/AzwmES>

It is not enough to place your code in an open setting. It is important to be active in the community. This helps indicate that the project is active. It also encourages others to jump into the conversation, deploy, and contribute.

# Break into libraries, microservices

Questions? Examples? <https://goo.gl/AzwmES>

Instead of open sourcing your big project, consider breaking it into multiple projects. This can be taken to an extreme, but it can be really helpful when done right. As we developed additional functionality for uPortal, we found ourselves developing microservices that would be useful in other projects as well. By creating these as separate projects, it was easier for other groups to start using that particular service without having to download uPortal.

# Use semantic versioning

Questions? Examples? <https://goo.gl/AzwmES>

As more people adopt your software, it can be challenging for them to keep current with the latest versions. You want to give them the flexibility of determining when they upgrade and semantic versioning gives them a clear indication as to what is a patch release, a minor release, or a major release and what level of pain will be inflicted on them.

# Expect to do 99% of the work, then be pleasantly surprised

Questions? Examples? <https://goo.gl/AzwmES>

Finally, don't think you can simply make your code open source and then have the world doing your enhancements and bug fixes. It is most likely that you will be doing all the work, at least initially. With luck, you have developed something useful and a community will start to grow around it. That has been the case with our latest front end we developed for uPortal. Surprisingly, even though no one was yet using our work, a young eager beaver employee of a private company was interested and started submitting pull requests that improved the project yet were things that were really low priority for us - but we eagerly accepted the contributions!

# Try it!

Questions? Examples? <https://goo.gl/AzwmES>

I hope this has convinced you to try open sourcing a project or adopting an existing project and getting involved in its development.

Questions?  
Examples?

<https://goo.gl/AzwmES>

info@my.wisc.edu

Questions? Examples? <https://goo.gl/AzwmES>

While we didn't have time for Q&A today, do add your thoughts to this google doc and we can use that as a springboard for further collaboration. It also contains a link to the google slides version of this talk with speaker notes and some resource links.

## Policy resources

- [UW System Administrative Policy 190 Computer Software Ownership](#)
- [UW System Administrative Policy 191 Copyrightable Instructions Materials Ownership, Use, and Control](#)
- [UW System Legal Topic Copyright](#)
- [WARF FAQ](#)
- [Legal Affairs](#)
- Ben Balter on [notices](#) and [CLAs](#)
- [ChooseALicense.com](#)

Link to the google doc version of this presentation: <https://goo.gl/Nk2ZBt>

Questions? Examples? <https://goo.gl/AzwmES>