Request for Proposals
The Python Software Foundation

Redesign and Implementation of Python.org

Submitted by:

Project Evolution
Providence, RI
www.projectevolution.com
Seth Burke | 401.270.4111

Revolution Systems
Lawrence, KS
www.revsys.com
Frank Wiles | 800.647.6298

Main Contact person for this Proposal:
Seth Burke – 401.270.4111
seth@projectevolution.com

Introduction

Project Evolution (PE) and Revolution System (RevSys) are pleased to present our submission in answer to the Python.org Request for Proposals. Project Evolution is a Python/Django-based design driven development shop with over 10 years of experience in the field of design, information architecture, user interface and content management systems serving Fortune 1000 companies. RevSys is a high-end development shop headed by Frank Wiles and Jacob Kaplan-Moss, both well-respected members of the open source community.

PE and RevSys have collaborated on projects previously of varying sizes and requirements. and make a strong design and development team. Together, we make a strong team for the front- and back-end needs of Python.org.

Our Philosophy and Process

Our Design Philosophy

Project Evolution is a design-driven technology provider. With that philosophy in mind, we deliver design solutions for a wide variety of clientele, from fashion companies to defense contractors to tattoo shops. Our approach for each is the same – deliver a design that is unique for that client, that places them in their industry but breaks them away from the pack.

This means that we do not have a cookie-cutter, template-factory approach to our clients needs. While we employ tools and frameworks that make web development more efficient, we do not squeeze a client's website into a predetermined box according to their industry niche. Each site is custom in its own right, and often the result of many conversations where we try to get to the heart of the client's message and what makes them special.

The Python.org Request for Proposals does a good job of initially outlining the site's main audiences and uses. Through additional conversations we feel that we can deliver a site wireframe and final design styles appropriate to the site's user base. The ideas presented here are for initial review and may change after we have more conversations with your redesign group and stakeholders.

Our process: User Interface Planning and Design

It should go without saying that the site should be intuitive for a user to navigate and leverage the full benefit of the tools and information provided to them. Content should be easy to read, the graphics should convey value and professionalism while also defining the brand. There should always be multiple ways for a user to navigate to various areas as everyone has a different way of working.

Project Evolution's planning and design process starts with a clear set of project goals. Visual design is used to organize the various parts of a site and enhance the client's brand, as well as to evoke a unique user experience. Planning and architecture supports the design and functionality of the site, while ensuring a fluid and scalable armature for the site to adhere to.

The stages of Planning and Design, including Approval stages and rough timeline, are as follows (Week numbers indicate that by the end of # week, this line item should be complete):

★ **Week 0** = Project Scope and preliminary budget
  ☆ Initial Budget Approval required
★ **Week 4** = Project Discovery, preferably led by an in-person kickoff meeting
★ **Week 6** = Planning Documentation – a refined Scope of Work and budget
  ☆ Revised Budget Approval required
★ **Week 12** = Design and Architecture – wireframes, user stories, design samples and Style Tiles
  ☆ Design Approval required
★ **Week 20** = Development
★ **Week 22** = Testing, Approval and Delivery
★ **Week 24** = Review and Assessment

I. Project Scope and Preliminary Budget

This document is PE and RevSys's first attempt at putting these items in writing. It constitutes a rough project outline before a proper Discovery Phase can begin.

II. Project Discovery and Definition

The intent of Discovery is not only to learn as much as we can about the project at hand, but also to learn as much as we can about your organization, your targeted user groups (audience), and where the product fits in the marketplace. We already have some idea, being very familiar with Python and having read the fairly extensive RFP, but we would still like to use this opportunity to hear from the board and other stakeholders about the goals of the project.

The Project Scope will be refined and the budget will be revised if the Discovery process raises issues or features not addressed in the preliminary scope. Opportunities to push these new features off into additional phases or redefine which features are absolutely necessary for Version 1.0 are also presented.

III. Design and Architecture

This phase will take the plans done in Discovery and flesh them out into full user stories, simple working wireframes and design style tiles.

User Stories describe the features that the system needs to support. They take the form of "This type of user needs to accomplish X, Y, and Z". These are typically specific functions that can be easily made into a task for a developer. The user story does not need to take into account how something works, but rather, what a function helps a user of the site accomplish.

Another name for these discrete functions is a **Minimum Marketable Feature** – a unit of business value of the project. The MMF is the smallest piece of functionality that will be demonstrated for the client and be ready for testing. MMFs are useful to focus on the core of a task or feature. In this way, we can be sure to define the core features of a product before getting caught up in the various interface niceties – animations, user feedback, or other programming methods – we can add to make it function better.

Wireframes are intended to be a first step that outlines navigation and the user workflow through a site or feature. They are simply skeletons without any design information to get in the way of how a site or feature needs to function. When appropriate, wireframes will simulate the desktop experience and also a smart phone experience if user flow needs to change dramatically.

Style Tiles are representations of the look and feel of a site without the skeleton underneath – they are a means of visualizing the site without getting caught in the weeds of the functionality. A style tile also gives the client an idea of what the site may look like independent of the device being used to view it – as web-capable devices are flooding the market, the need for a more flexible design is becoming more and more important.

IV. Development

The development process is where the rubber meets the road. If all was planned, scoped and wire-framed correctly during the previous phases, the development of a site will have clear metrics to measure success.

Our standard back-end software stack includes the Django web framework, gunicorn, Nginx web server/reverse proxy, PostgreSQL database, memcached, redis and Ubuntu 12.04 LTS. Other tools are used when and as necessary. The software stack necessary for Python.org may vary, though we're committed to using only tools licensed under open source licenses.

On the front-end, our standard tools and practices include the use of Zepto (a jQuery derivitive with better support for mobile/touch interactions) and Backbone.js as the basis for our JavaScript development, judicious use of the latest web standards (HTML5 and CSS3), and incorporation of generally-acknowledged best practices for front-end engineering. Our CSS is complied with SCSS and Compass for more authoring flexibility and output configuration.

V. Testing, Approval and Delivery

While in development, a staging server is deployed that is identical in configuration to the eventual production site, so that prior to launch all aspects of the site can be tested in a truly comparable environment. **The client is required to create a group of users that will have access to this server for testing and general feature and design review. Sign off from this user group is required for changes/features to be pushed to production.** Considering the wide array of use types that visit Python.org, this group of test users is extremely important for providing a consistent user experience for all users.

User stories can now be used to test and see if the product we are developing has the functionality that was originally intended. In this way, success can be clearly measured.

Once a deployment has been approved by the client, the feature, improvement, or entire site is pushed to live production servers. These servers are monitored for performance issues and server-side or core code changes may be suggested based on this real time data. The beauty of our agile method is that it is built around continued development and improvement of a product allowing unique features to be developed and released in shorter timelines.

For this type of site – one that garners much community support and one for whom the audience probably has a high "early-adoption" rate – we suggest staging an open development version of the site, perhaps at new.python.org. Mechanisms would be put in place to collect user feedback, ideas for new features, and generally give the community a chance to buy-in and be a part of the evolving Python.org. This site could be used to stage releases throughout the life cycle of the site, and always be a beta testing ground before new features get pushed to the base production site. *This idea is open to further discussion and may not be part of the initial cost estimate.*

VI. Review and Assessment

After a live deployment, is it our suggestion that the client and Project Evolution/ RevSys teams sit back for an assessment of the outcome. This assessment can take many forms, but the idea is to review the workflow of the product as well as the workflow of the project and to set the next phase of goals, features and/or improvements to the site.

If new.python.org is a part of the new site (or released as a post-launch feature) a workflow should be put in place to review comments, vet good ideas as well as system bugs, and put them through a support workflow either with our team, with a team of active community members, or a hybrid group of participants.

Our Suggestions for the Redesign

To each point in the Purpose section of the RFP we offer these suggestions:

The next implementation of the site should:

Boast a modern design and experience

Our version of a "modern experience" will vary from others, we are sure. More can be discussed during the Discovery phase of the project, but in general, here are some ways that our style may be described: direct, clean, subtle, organized, and uncluttered. We have dabbled in some of the more recent trends, but in the end we gravitate towards the simplest solutions.

We also find that while CSS3 effects are nice, too many can make page rendering on a mobile device sluggish, so we temper our use of these effects in favor of visual simplicity. Again, it can vary from project to project, but the goal is to have a stylish site that reflects the brand and is a quick download or render on any device.

Our design philosophy is future friendly – the best practices in place now will be ready for the devices that have yet to be invented. This includes scalable frameworks, iterative development, responsive design and progressive feature enhancement. We support a mobile-first approach for our front-end framework and integrate best-practices for CSS and HTML5 development with inspiration from HTML5 Boilerplate and other sources.

Browser Support

Progressive enhancement dictates that older browsers should not have broken experiences, but that graceful degradation is allowed to occur. Our baseline of browser testing is as follows. Any additional support will require more development hours for testing and bug squashing:

★ Internet Explorer: 8, 9, 10 (when available)
★ Firefox: 3.6.25 (latest 3 series) and the auto-updated latest release, currently 9.0.1
★ Chrome: latest auto-updated release, currently 20.0.1132
★ Safari: latest 5.1 release, iOS 4 and up
★ Android: version 2.3 (Gingerbread) and newer

Additional browser support for desktop and mobile devices can be discussed during the Discovery phase to ensure that the correct browsers are supported based on your user analytics. For example, no specific requirements are in place for Linux or Blackberry users, but Python.org may see more traffic from these devices than we normally see, and therefore a test plan needs to be supported for one or two of the popular Linux browsers (Firefox and Chromium, probably) and a few of the widely used Blackberry browsers. *Additional time for testing on these devices should be considered along with whether or not support for some user groups needs to be part of the initial re-launch.*

Javascript Support

While modern browsers and devices support Javascript, we realize that even if 1% of users/devices do not support Javascript, that could still be 150,000 visitors (assuming 1.5 million monthly visits).

With a tool like Moderizr, Javascript support is detected in all of our recent website deployments. Without javascript, a “no-js” class on the HTML tag is used to target CSS styles for those users. We test the site without javascript enabled to ensure that functionality is not broken, and we use CSS transforms and transitions whenever possible, as they gracefully degrade better and keep the Javascript load light.

Navigation menus that use drop-down effects are strictly CSS, which ensure that they work in modern browsers and devices. Further, we use HTML5 form elements which offers browser validation, and adds a fallback level of validation when javascript is not present.

Provide concise and intuitive navigation

Navigation of a site on this scale will always be a challenge. Once a new site map has been established, it will still need to be flexible enough to accommodate future growth.

We think that Python.org has three main groups that it needs to talk to: the Layperson, the IT Manager, and the Developer. The current site does a good job of supporting the developer who already knows Python, and the new site needs to address the questions that a Layperson may have as well as addressing the concerns of IT Managers who are interested in adding Python to their configurations (new content may need to be crafted for this audience). On top of this layer are the standard About, Foundation, News and Download sections that need to be accessed.

From the RFP, we know that the user groups that have been identified are:

★ Potential Users
★ Existing Users
★ Users of Other Languages
★ Managers
★ Contributors
★ Sponsors

We see these breaking down into our proposed three tiers as such:

★ **Laypersons** – Potential Users, Casual and "accidental" browsers
★ **IT Managers** – Users of Other Languages, Managers, Sponsors
★ **Developers** – Existing Users, Contributors, Sponsors

Additionally, we suggest a set of Global Navigation that would be omnipresent on the site to showcase links relevant to all groups, such as:

Download | About Python | About the Foundation | Fund Python | Python News

The Main navigation could consist of "Super Navs" with CSS dropdowns in these three areas:

For the Layperson

★ What is Python and how is it used?
★ Who uses Python (Quotes and Success Stories)
★ Beginner's Guide/Non-Programmers
★ FAQs (low level)
★ etc...

For the IT Manager

★ Python versions and application packages
★ Security, Scalability, Legacy System Integration and Systems Administration
★ Who uses Python (Quotes and Success Stories)
★ FAQs (high level)
★ etc...

For the Developer

★ Current Documentation
★ docs.python.org
★ Python Wiki
★ Python Insider Blog
★ Python 2 or 3?
★ Python Planet: Other Python libraries
★ Audio/Visual Talks, Screencasts and Slideshows
★ Non-english documentation
★ Contribute to the Python codebase
★ etc...

Not all pages currently in the navigation are represented here. We think that this might be a perfect opportunity to move some pages around, consolidate, and generally review current content and its organization.

The footer is currently neglected by the website, and we suggest that it should be leveraged to present an organized site map (Super Footer). In this way, it should be easy for a user new to the site to find what they need.

Wireframe sketches of a potential navigational structure are attached to this proposal (at the end of the document). These wireframes are preliminary only, and were done in advance of having any conversations with the PSF.

Showcase the simplicity and elegance of the language

There are a number of ways that this can be accomplished. Below are some of our initial ideas, and we are open to more discussion about this topic of evangelizing to new users:

★ Explain the tenets and Zen of Python, with credit where credit might be due (Mark Pilgrim): Python is such a readable language, that we can learn it by reading well-written Python programs.
★ A simple "Getting Started" page could highlight a few common functions and how they might be written out in other popular languages (PHP, Ruby, Perl), and how that compares with how it might be written in Python.
★ Video tutorials or screencasts of how one might get a simple project started in Python
★ Videos or screencasts of someone solving a complex task with Python. Meant as a way to communicate how it can handle complexity, without worrying about a novice user understanding it completely
★ A live, interactive Python interpreter, ala try.github.com or tryruby.org. Open to further discussion and technical review, but the general consensus among PE and RevSYS is that if possible, it would be a great tool and a great way to show users new to the language how elegant it is. *This potential tool is not covered in the current cost estimate.*

Attract and convert potential Python users and Python Software Foundation sponsors

The redefined navigation along levels of technical expertise should go a long way in converting and convincing the right parties that Python is the language for their project. Gathering additional donations can be a little tricker.

Generally, it follows that a well designed site that highlights and supports the existing developer community will foster more involvement from businesses, charitable endowments and individuals. Moving the Donation link to a more prominent position may also help, though we understand that any Donation link should be tastefully executed and not detract from the strong community that is already present.

We would like to surface the Success Stories to become more prominent, particularly on the homepage and as part of the messaging (sidebar) for the Layperson and IT Manager templates. These stories are currently buried on the site, which is

a shame, as they provide a great means of telling the story of how a client came to use Python or which well-known projects are running on Python code. Many of the companies featured in these stories are household names, and could go a long way in convincing would-be programmers and IT Managers.

A submission form for new stories should also be prominent. These submissions should go through a review/moderation process, but the idea here is to simply make it easier to tell Python.org how awesome Python is for an organization.

We would also like to come up with ways of surfacing the list of Donors in a non-preferential way. Highlighting the community that supports Python is a great way to "softly" solicit for additional funding support.

Represent the vibrant, active community

The Python community is indeed an active one, and there are many more ways in which we could help make this apparent. We plan to incorporate as much community activity as possible by pulling in community feeds such as:

★ RSS feeds (i.e. "planet"-style blog feeds)
★ Twitter activity (from prominent Pythonistas, #python, etc.)
★ Upcoming events gleaned from Python user groups, meetups, etc.
★ Python-related development activity on Github, Bitbucket, Google Code, etc.
★ … and any other types of community activity that come up during Discovery

In all of these cases, this community content will be readable and writable via REST APIs such that the community can contribute their own content. These APIs will be well-documented (with example Python code, of course!)

Make it easy for a wide range of contributors to add content

For this project we believe that the Django framework is a no-brainer for setting up basic content editing capabilities. The built-in Django Admin features are well suited for the content management of this site, and when features need to be modified or simplified, Project Evolution has experience with modifying Django Admin to streamline the workflow or support custom features and content types.

Like the community content, all of the admin-contributed content will be available via REST APIs (access-controlled as necessary). Among other uses, this will allow us to create tools to allow off-line editing of the site's content.

Enhance the visibility of the PSF and its sponsors

Some of the ways in which we can highlight the Foundation and its sponsors have been discussed by other points. Other ideas include a clearer delineation between where Python the language ends and the Python Foundation begins.

It could be that it is time for the Foundation to have a web presence of its own – a sister site to Python.org. If that is too far out of the question, than we suggest that the Foundation section be designed more as a subdomain with its own set of slightly different templates (perhaps foundation.python.org). This would allow things like a footer of sponsor logos (randomly rotated or available to a certain tier of sponsorship) to be active in the foundation section and not impede upon the rest of the site. This could also give the Foundation more of its own brand. Allowing the rest of the Python site to stand on its own would make both entities stronger.

Provide examples of Success Stories

Success stories (as mentioned in *"Attract and convert potential Python users and Python Software Foundation sponsors"*) are a powerful way to tell the Python story. At this stage we believe that surfacing these stories more prominently on the homepage as well as sprinkling them throughout the site in a sidebar widget or in the footer can help make them easier to explore. An excerpt should be crafted for each one that makes it easy to digest and hooks the reader into exploring more.

Enhance the visibility of alternate implementations (Jython, IronPython, PyPy...)

One of Python's real strengths is the availability of different implementations for different platforms (and purposes). We'd like to explicitly call out this fact. This could link to an expanded "Which Python is for me?" page explaining the differences and helping users decide. This might be especially helpful for deciding between version 2 and 3, as well.

We'll need assistance from the PSF and Python community in drafting this content and getting it right. This idea could be part of the Python Community, or live under its own name, perhaps something like the "Python Ecosystem".

The community pages can also draw in activity across all Python implementations, thus making those communities' progress and developments more visible.

Exist atop stable and scalable infrastructure (99.99% availability; able to survive a slashdotting).

RevSys has extensive experience designing, building, and maintaining fast, stable, scalable infrastructure – it's literally our bread and butter. Scaling Python.org will be fairly straightforward. The majority of the content is highly cacheable, so we plan to leverage a front-end cache, most likely Varnish, which will take the majority of the load. We design systems with redundancies at each layer (multiple web servers, multiple cache servers, etc.), ensuring that there isn't a single point of failure. We'll work with the Python.org infrastructure team to design a system that's scalable, robust, and fast. We plan to performance test throughout and make sure we're developing a site that's rock solid.

According to the current Python.org's webstats (www.python.org/webstats), the site gets roughly 1.3 million hits per day, or an average of around 15 requests per second. This is well within the bounds of what RevSys has worked with before, and no problems handling the site's traffic are currently anticipated.

Additional Ideas (not addressed by the RFP specifically)

The ideas outlined below are open to further discussion and the current estimate of costs does not take these into account.

Video

While video is difficult to prepare, it is an effective means of communication, and often will engage a different type of user than the site currently. A new site with a new framework should have the ability to leverage video content. We imagine having the ability to present tutorials, expert-level "tips and tricks", screencasts, and testimonials, as well as the archived conference material available from years of Python conferences. We'd like to integrate with http://pyvideo.org and bring this content scross in a similar manner to other community content discussed above.

Python Meet Ups (http://python.meetup.com)
Regional Python MeetUp groups are another great source of content and community support. Python.org could provide tools for these groups to collaborate on large-scale projects, become test subjects for new Python features, and generally crowd source and publicize their local efforts to a worldwide audience. Further, MeetUp.com charges group organizers – a free service offered to Python MeetUp groups would be enticing and a great way to get even more community involvement with the language.

These are some of the "wouldn't it be great if..." ideas we would like to discuss further for a Phase 2 (and beyond) of the site, once a new framework is in place and accepted by the community.

Qualifications

Information Architecture

Information architecture is the first step towards a great site. Our planning process puts this at the front of the project, as it lays down the skeleton that the rest of the site adheres to. PE has experience with large sites consisting of many levels of navigation. PE also has experience with user segmentation, which provides specific navigation according to a user's place in the company hierarchy. While this is not a tool that the Python.org site will need to use, it is worth noting as an excersize in IA/UI design – an IA that responds to the user's heirarchy and what they need access to.

Our initial thoughts about IA and UI are outlined in other portions of this RFP. In general, this is a conversation that needs to be had early with the client stakeholders to ensure that all navigational elements are on the table and discussed. In this case, we think the User Groups that we end up defining will help divide the content into manageable chunks according to what these users may most likely be looking to accomplish.

Wireframe sketches of potential IA are attached to this proposal (at the end of the document). These wireframes are preliminary only, and were done in advance of having any conversations with the PSF.

Visual Design and UI/UE

Project Evolution has designed sites for many large corporate and community entities, and often we find that new clients are attracted by our approach to simple, modern UI. An emphasis on a clear navigational hierarchy leads the design, with color and typography that extends the clients brand supporting the usability of the site. Mobile-first methodologies ensure that this approach work for as many devices as possible.

RegisConnect.com is an employee portal site, and is used for employees at all levels in the company for training, task management, and communication. The site has a daunting level of content, but with employee segmentation, the navigation simplifies itself based on the user's position in the company hierarchy and their geographic location – the tools they need are available, and none of the content that does not apply to them is present. Since the site is a closed (sign-in) system, we were able to leverage what we knew about each employee to make this possible. The design is colorful, full of iconography, and extends Regis's existing brand standards to help employees find the tools they need, even though the navigation is already limited by segmentation. The interface for these tools is as simple and unintimidating as possible, which boosts employee acceptance and productivity.

The Westport, Connecticut school district hired PE to redesign the entire network of 8 schools, starting with the main District site. This framework was designed with a rollout for the individual schools in mind, and will provide a consistent experience for parents and students as they graduate from elementary to middle school and onto high school.

In all cases, PE has taken an existing brand and expanded upon it for a greater online presence. For many clients, our refinements have led to a more consistent brand that has been adopted for other materials and ephemera.

Front- and Back-end Development

Project Evolution and RevSys offer a strong knowledge-base for back-end development in Python, Django and PostgreSQL. Both teams have extensive experience managing large database-driven sites with solid results.

RevSys and its employees have an extensive amount of experience with Django as an open source project, but as a company we are often called on to do code reviews

of teams of all levels of experience and to help "fix" otherwise broken or inefficient development methodologies. We plan the build a technical foundation that's efficient and easy to use. The code for python.org will be open source, so we consider it part of our scope to make it easy for the community to contribute changes. This means things like making the site easy to run locally, giving the code great test coverage and an easy-to-run test suite, and including documentation about how the site works and how to contribute.

For front-end development, the Project Evolution team prefers new tools like SCSS and Compass to manage CSS programmatically. We are a small team, but our work is well documented to ensure that new members or third-party teams can jump in to manage code at any level of the site's development.

System Architecture and Monitoring

Related to our answer for *"Exist atop stable and scalable infrastructure..."*, we can say that these ideas are based in the real world structure of some of our larger, high-volume sites.

RevSys' main focus as a company is helping other Django users deploy, tune, and scale their applications and infrastructure. While RevSys does often work with clients in a development role, dealing with obscenely high traffic sites is what we are known for in the industry. Terabytes of databases and millions of users per day are things we are very comfortable dealing with.

RegisConnect, an internal employee portal for Regis Corporation (PE's largest client, and one whom RevSys has contracted to work on modules for) handles up to 50,000 employees daily. A high-volume month sees 20,000 unique visitors accounting for 92,000 visits and 402,000 page views. A heavy-traffic day can see 22,000 page views, which is significant given the content segmentation and business rules required. Many levels of employee hierarchy segment the content, delivering the navigation that each employee needs at their job position and geographic location.

A team of PE employees handles monitoring and a Help Desk during business hours. One developer is on-call during off business hours to handle any problems that arise. Hosted Chef handles provisioning services, while Sentry and Tracelytics help handle performance analytics and helps us tweak the configuration as needed.

About the Applicants

RevSys

Revolution Systems, LLC., based in Lawrence, Kansas, was formed in 2002 by Frank Wiles to help businesses benefit from open source software. While many large organizations use open source software internally (sometimes without their knowledge), he realized that many organizations did not know how to properly take advantage of this revolutionary type of software.

Jacob Kaplan-Moss joined in 2009 as an experienced software developer with a focus on web application and content management architecture. Jacob is one of the lead developers and co-creator/founder of Django.

Revolution Systems' primary focus is building and maintaining customized web applications, consulting, and technical support for a variety of open source technologies with a focus on Django, PostgreSQL, and performance/scaling related consulting. Clients include the Discovery Channel, Cox Media Group, RueLaLa, National Geographic, Votizen.com, EveryBlock.com, Scottrade.com, Threadless.com and USA Today, to name a few.

Both Frank and Jacob would be available for this project, and will be involved in much of the initial planning and Discovery as well as back-end architecture, development, and performance tuning.

While most of RevSys' work is behind the scenes, below are a few publicly accessible sites we've been a part of to a significant degree:

www.ljworld.com, www.lawrence.com, www.kusports.com – Three content heavy properties owned by the World Company, the company that created Django and the first significant Django CMS, Ellington (www.ellingtoncms.com). Incidentally, Lawrence.com was the first Django website ever.

www.cashstar.com – CashStar is one of the largest digital gift card providers in the world servicing clients such as Gap, Home Depot, Pottery Barn and many other large brands. We were brought in to help redesign and rewrite the customer experience and move their infrastructure to the cloud to handle the busy holiday buying seasons without overloading their back office systems and losing sales.

http://www.khi.org – The Kansas Health Institute. We help support this site and have made some customizations to the Ellington CMS used by KHI.

RevSys' team includes:
Frank Wiles, President and Founder
Jacob Kaplan-Moss, Partner
Jeff Triplett, Engineer
Jacob Burch, Engineer

References
Jeremy Dunck | Votizen.com jdunck@votizen.com
Evan Walsh | Ruelala.com ewalsh@ruelala.com
Mike Shields | KHI.org mshields@khi.org

Project Evolution

Project Evolution (PE) is a design driven development team founded in 1999 with clients ranging from school districts to Fortune 1000 fashion conglomerates with international holdings. The 12-person team includes creative leads, front-end and back-end developers and associated support staff all committed to open-source technology..

PE's projects all revolve around a custom CMS framework based on the Django platform. Our framework is designed with the novice user in mind, as many of our clients are non-technical marketing or HR staff. The strong emphasis on usability and simplicity makes our CMS one of the easiest to train clients on, and their productivity is part of our system's deliverable.

PE's two lead designers have 12+ years of experience (each), with expertise in information architecture and user interface design for complex web software systems. The developers and designers at PE have a singular goal: great-looking sites and powerful performance.

For this project, we expect to have both design leads involved in the IA of the project, while one will take the lead during UI. Two front-end developers, our Technology Director, and other developers as needed will be involved, as well as the team at RevSys for the back-end infrastructure. Project managers, team support and accounting will be involved as needed.

While PE's projects are wide-ranging, we do not have a site that closely mirrors the site outlined in the RFP – a large community-oriented site with vast amounts of technical documentation, as well as information for the layperson, events, and case studies. What we hope to highlight in this proposal is our method of thinking and working through a new site or software tool, which we think can be applied to any type of project of any size. Design is then laid on top of a well-structured framework, and must reflect the client's brand and audience. Links below are for reference, therefore, and do not reflect what we would build for Python.org:

Clients we have worked with whose sites are protected by a login or NDA: Lockheed Martin, Microsoft (Windows 7), Johnson & Johnson, Bristol Myers Squibb, Farmington Bank, and the Regis Corporation.

Regis Corporation Employee Portal timeline (with screenshots)
/www.hrlovespe.com/timeline.html

First Data Employee Benefits www.firstdatabenefits.us
Westport Public Schools www.westport.k12.ct.us
PGAL Architects http://www.pgal.com

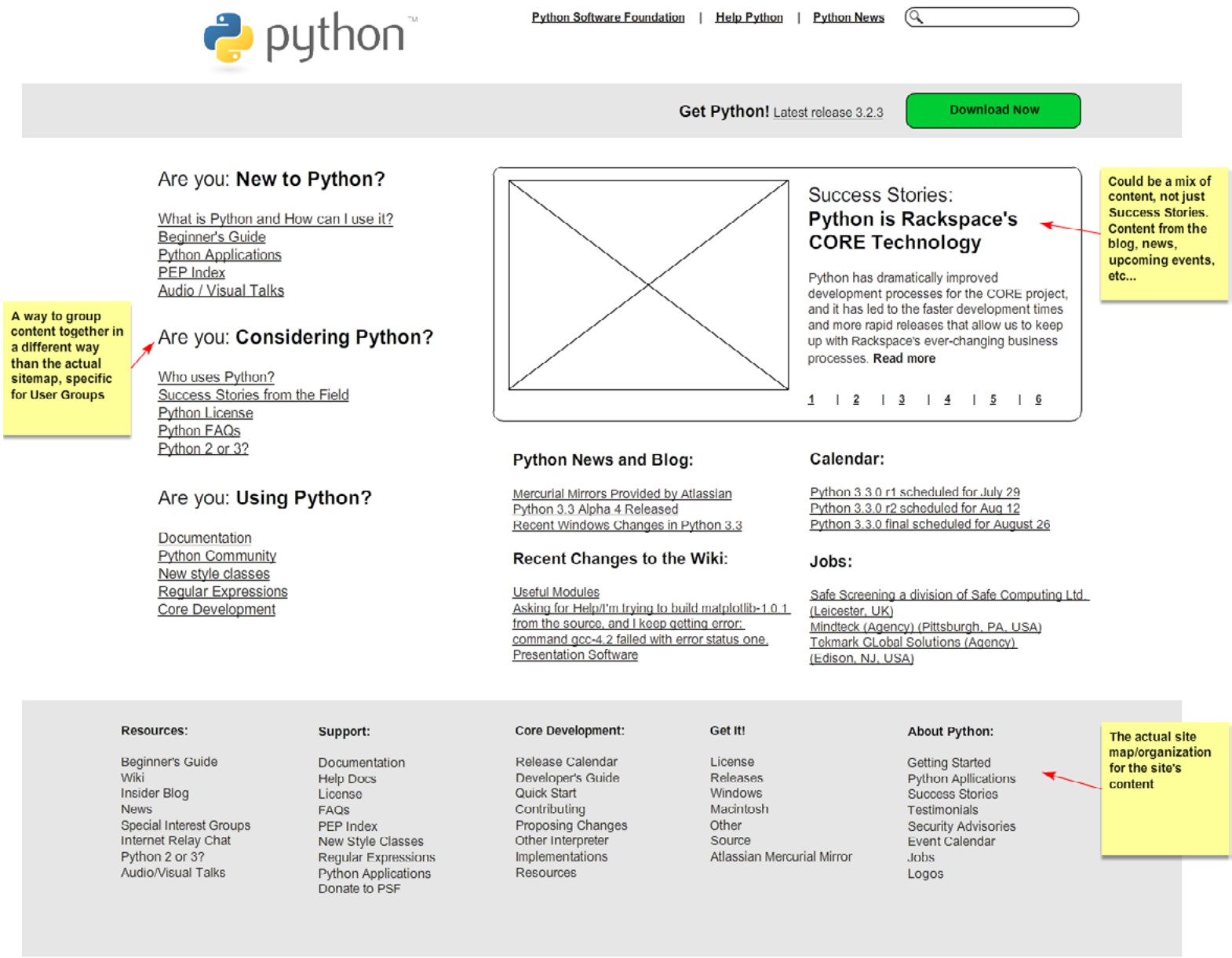
Project Evolution's team includes:
Seth Burke, Founding Partner and Creative Director
Winnie Tantraporn, Founding Partner and Business Director
J. Hogue, Design Director
Peter Landry, Technology Director
James Re, Front-end Developer
Barry Menard, Front-end Developer
Ray Schamp, Back-end Developer
Victor Noagbodji, Back-end Developer
Jarrett McPhee, Designer and Customer Support
James Caruso and Chelsea Flynn, Project Management and Support
Faith Zeller, Sales and Marketing Director

References:
Allison Brown | Regis Corp.
952-806-2169 | allison.brown@regiscorp.com

Gay Copeland | First Data Corporation
402.222.7440 | Gay.Copeland@firstdata.com

Natalie Carrignan | Westport, CT Public Schools
203-341-1217 | Natalie_Carrignan@westport.k12.ct.us

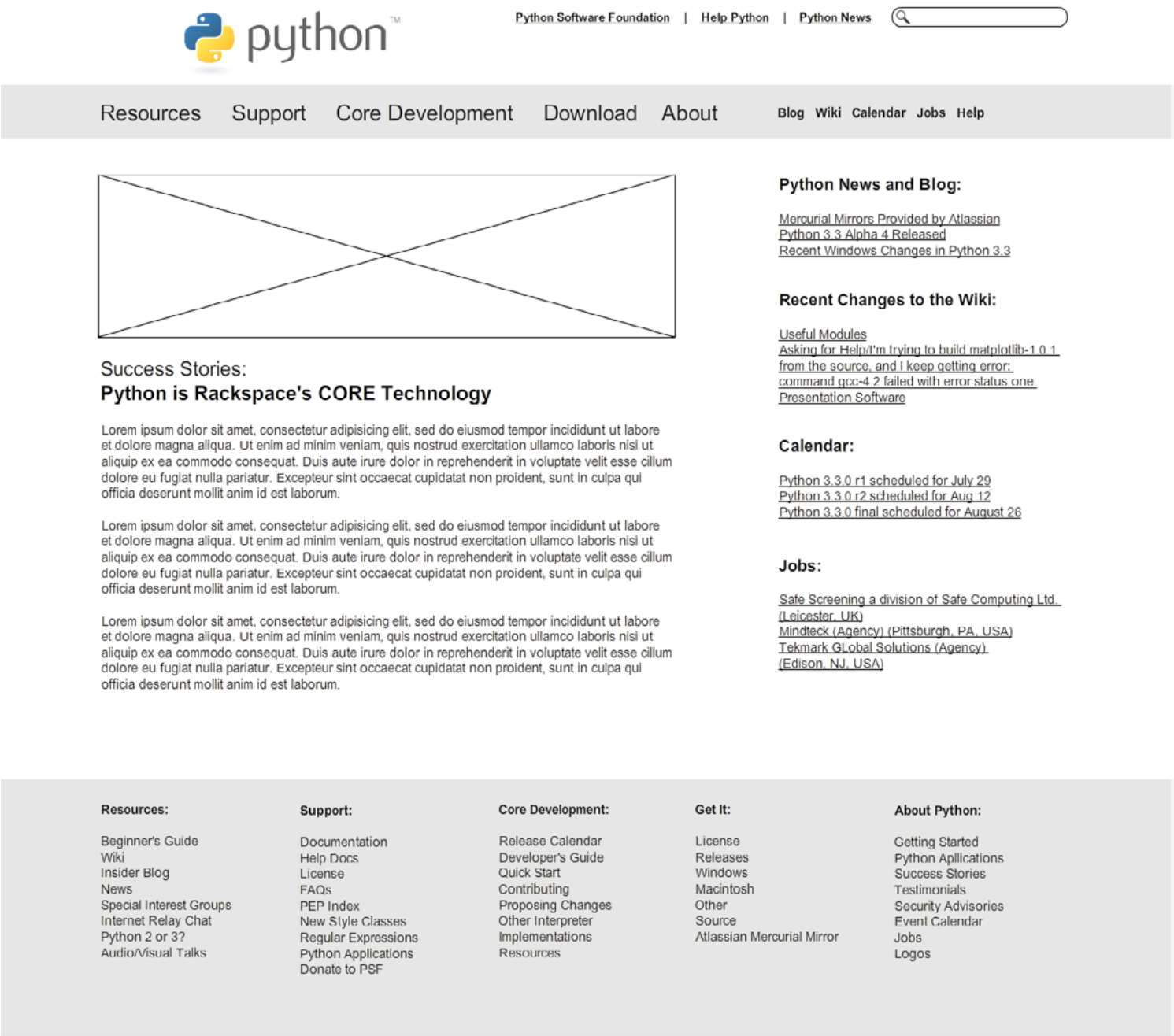
Wireframe Sample #1 Homepage



View these wireframes online:

http://app.mockflow.com/view.jsp?id=FE72754CAFD69A7982519BCEE5B58FE3

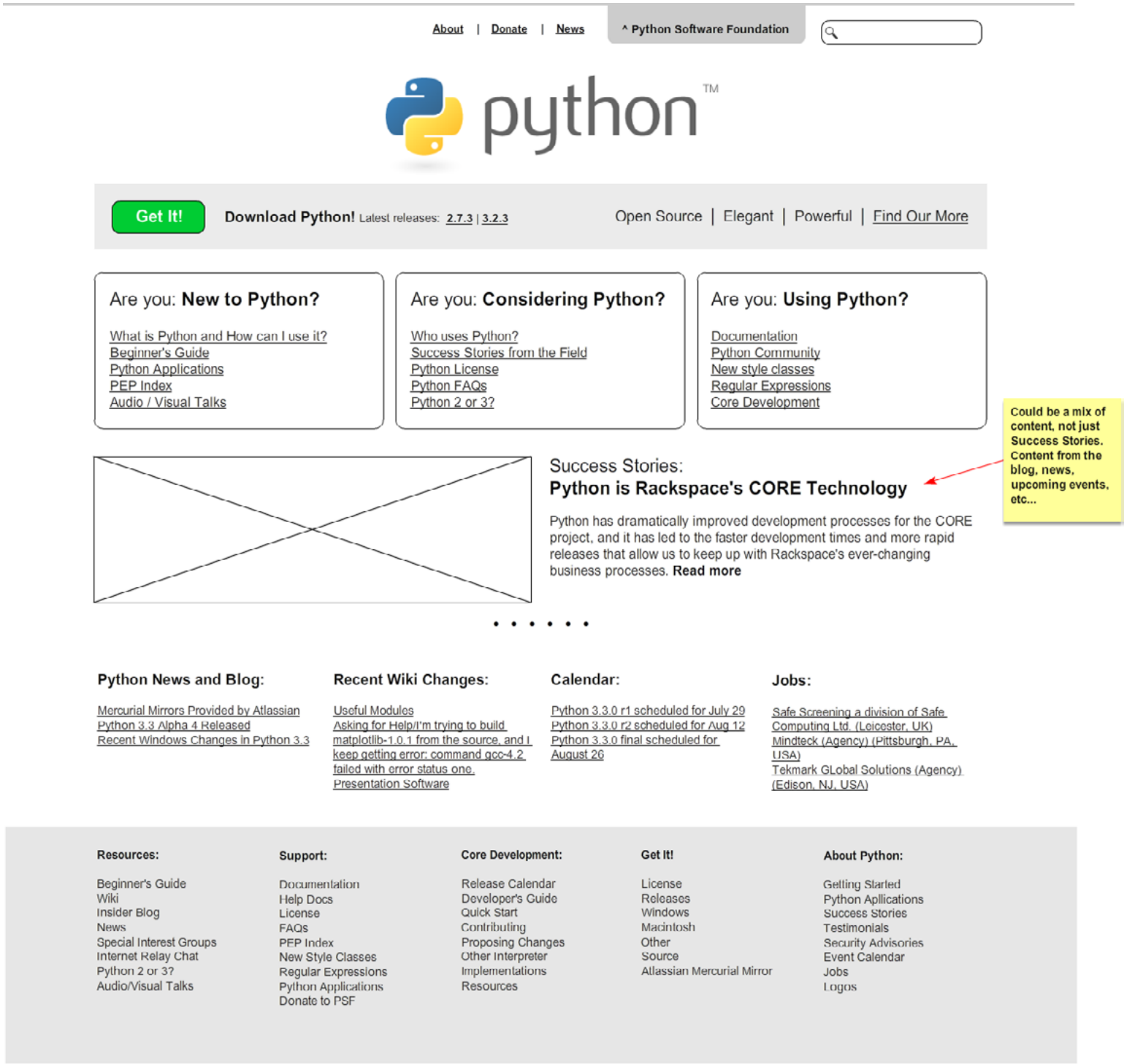
Wireframe Sample #1 Inner page



View these wireframes online:

http://app.mockflow.com/view.jsp?id=FE72754CAFD69A7982519BCEE5B58FE3

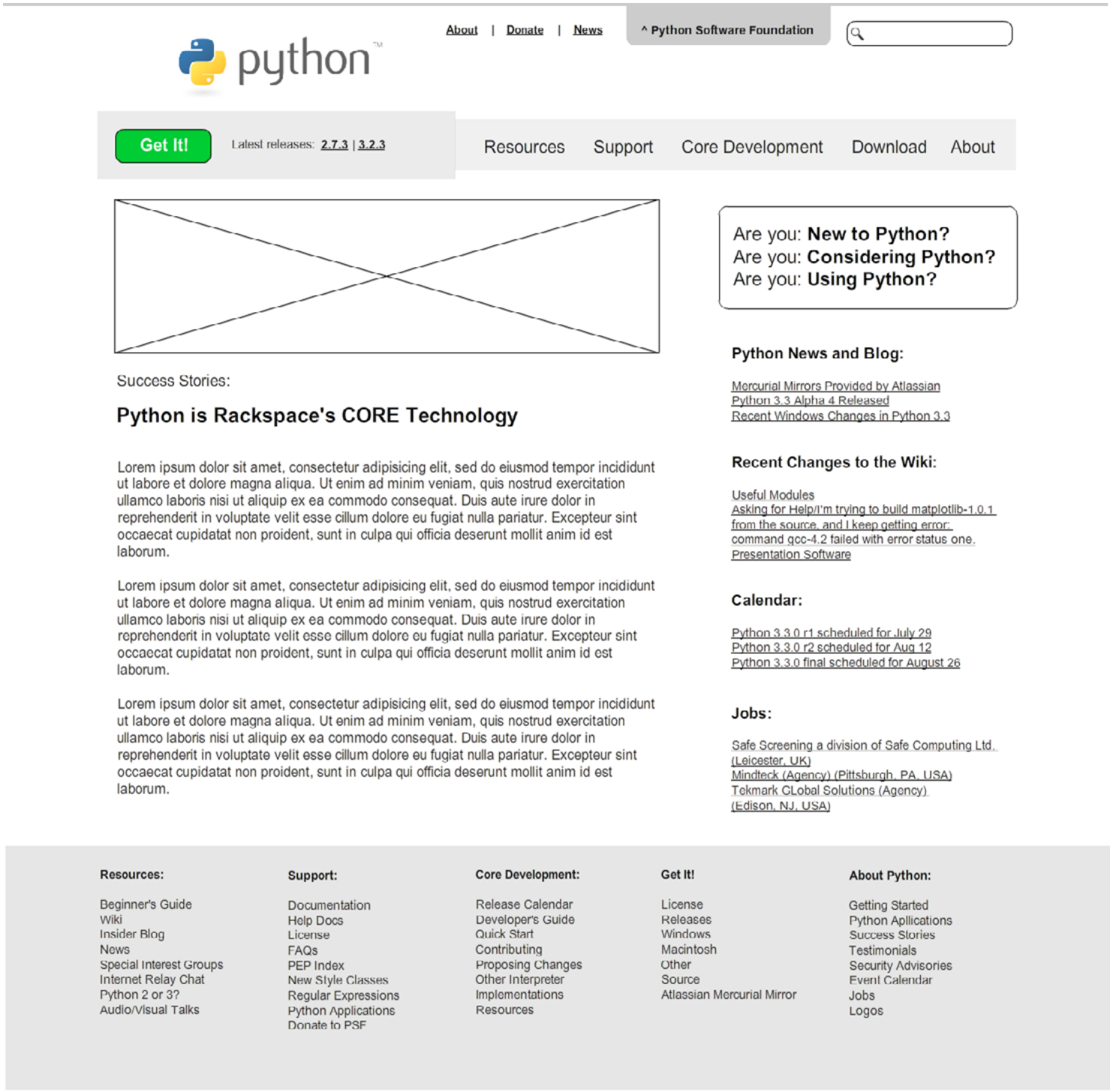
Wireframe Sample #2 Homepage



View these wireframes online:

http://app.mockflow.com/view.jsp?id=FE72754CAFD69A7982519BCEE5B58FE3

Wireframe Sample #2 Inner page



View these wireframes online:

http://app.mockflow.com/view.jsp?id=FE72754CAFD69A7982519BCEE5B58FE3

Updates and Answers to specific PSF Board questions

Question 1 from the PSF

RE: the interactive Python demo. "This potential tool is not covered in the current cost estimate.". We would like, if possible, to get some numbers in regards to this feature. Note that http://try-python.appspot.com already exists and could be potentially licensed/leveraged.

We consider this feature as "nice to have". We are not sure if it is feasible given the traffic that **python.org** gets, and we're not sure if it will necessarily make sense in the design visually. If we can make it work, however – both technically and aesthetically – we're going to.

There would be no extra cost associated with this tool: it is either something that will work, and will fit into the budget we've proposed, or it's something that simply can't be done and will be skipped. It does not hinder the overall goal of the **python.org** site if this idea is not in place for the final launch.

The board is correct in that there are many similar tools in the wild. If it makes sense and if there is an existing tool with the right amount of flexibility and power, than we can certainly look into repurposing it to meet the current needs.

Question 2 from the PSF

Can you please discuss the outlined requirements in the RFP surrounding offline/online content editing and workflows. It is important that the system architecture support both the online content editing and offline (e.g. checkout) model of content editing. We did not see clearly defined points addressing this requirement.

Please excuse us for this oversight. We had discussed our solution internally, but it did not make it into the final proposal document.

Our plan is that all content – all the CMS content manageable through the online admin interface – will actually be backed by a Mercurial repository. Developers interested in offline and/or non-web-based editing will be able to check this repository out directly, make edits to flat files, and push that content back. This means that any action that can be performed through the web-based CMS can also be performed with command-line tools. We'll document this workflow as part of our process.

Additionally, we're planning comprehensive API coverage: every single bit of data on the site will be exposed via REST APIs. As with Mercurial, any action performed

via the CMS will have an API equivalent, and we'll document the REST API. This means that tools that encapsulate offline workflows could also be written, though we do not have plans to write any of these offline tools as part of this proposal, except perhaps as a demo script.

Question 3 from the PSF

Can you please discuss the conceptual architecture of the site given the editing needs? We understand Django is the framework of choice, but how does that play into the content editing, sub-site creation and other thoughts outlined in the RFP?

In essence, data on the site will be backed by traditional Django models, and also by a "content repository" – the Mercurial repository we mention above. Most tools (the public website itself, the CMS, the APIs, etc...) will deal with Django models in a traditional manner, but it will be possible to side-step Django and access the content repository directly.

Jacob put together a quick diagram to sketch out what he is picturing: https://www.evernote.com/shard/s25/sh/cc30a283-e434-48dd-ad52-1dc6cea83e8b/2cc518131c016c58597514671fb746e4

Hopefully that makes the ideas clear, but please let us know if we can add more detail.

Question 4 from the PSF

RE IA/UX: We have a second bid whose IA/UX mock ups and skill set would seem to work extremely well for the project; especially if married to your teams skill set. Would your team be willing to work with a second company/group on the IA/UX front if they were open to it?

After internal discussion and further discussion with representatives from the PSF, we have agreed to the following, subject to approval of Divio, the third-party UX team mentioned as part of this question:

★ The **PSF** agrees to pay **Divio** a licensing fee for the work put into the proposed UX mockups.
★ **Project Evolution** & **RevSYS** agree to receive those mockups as ideas that the **PSF** considers well suited for their aesthetic and design goals.
★ **Project Evolution** – as the design partner in this project – will interpret these ideas and presented our own set of proposed templates. The **PSF** board continues to have the final sign-off power of the proposed templates.
★ **Divio** agrees to be a "User Advocate" during the process of the project, and offer their expertise as a UX consultant when the project is ready for that kind of feedback before final release.

Both Project Evolution and RevSys are comfortable with this agreement and look forward to seeing the aforementioned mock-ups.

Question 5 from the PSF

Finally, the [board] has discussed the offer of in-kind trades, and have agreed that having the team's information on the footer and within the credits of the site are acceptable, and if desired other avenues of in-kind trades can be made.

We are sure that the PSF and PE/RevSys teams will be able to come up with something all parties are happy with. It is low-level concern for us at this stage.

Thank You

Thank you for reviewing our proposal. This is not just a redesign to the Project Evolution and RevSys teams, but a chance to give back to the Python community – a community that we are passionate about and already evangelize. The chance to make Python.org stronger and faster is a fantastic opportunity to help show the world how great the language is, and how elegant open source software can be. Once again, thanks for the opportunity.