CASE STUDY 1:

For a large non-profit organization established over 40 years ago with over 22,000 members, Up and Running Software (UAR) built a new application to serve both as their public-facing informational website and as their membership management and billing tool. The new application provides each of the dozens of chapters of the organization with a chapter homepage and tools to assist the chapter leadership with managing the chapter and their own membership.

- The application is based on the WordPress platform. It makes extensive use of existing WordPress plugins to provide a large amount of out-of-the-box functionality, however many of the plugins are extensively modified to align their behavior with the existing business processes of the organization. These modifications include:
    - Customizations to an event management plugin, event-manager:
        - Adding the ability to restrict registration to specific types of members
        - Adding an extensive system for allowing members to sign up for optional event workshops as part of the event registration process
        - Adding special business logic for giving automatic discounts to certain types of members
    - Customizations to a member management plugin:
        - Customizing the account address management features to support custom business logic for required address components and to allow for multiple addresses on an account
        - Changing account expiration logic to include a grace period and custom business logic for determining the start of an account’s membership period
    - Extension of the PODS plugin:
        - Modifying the image uploader to permit customizations as part of the process
        - Giving users the ability to draw a custom cropping region on the uploaded image for use as a thumbnail
- The application integrates with Authorize.net to provide automatic recurring billing for membership dues. New members are also able to sign up through the website.
- UAR constructed a search system to index and allow sub-second searching of a large number of member and member-related records. The search system makes use of MySQL's fulltext index type and uses a message queue system for indexing to throttle load on the server. Search pages on the main WordPress application interact with the search system via AJAX requests. A search system was designed to be independent of WordPress to avoid the performance overhead of invoking the WordPress framework on every AJAX request.
- UAR integrated the application with ConstantContact to facilitate the sending of bulk emails to subsets of the organization's membership. The code uses the ConstantContact API to send a list of member email addresses to ConstantContact. The application provides an interface for

filtering this list of members based on customized criteria such as a member's registration date and type. Once the list of member email addresses is synchronized to ConstantContact, the administrator can either use ConstantContact's website to send a mailing to the list, or they can use an interface built into the application to compose and send a message via the ConstantContact API.

- As part of the transition from the organization's legacy MSSQL-based application to the new LAMP-based application, UAR wrote migration code to transfer and remap the data from their old database into the WordPress database. This transfer posed several challenges, including:
  - Dealing with encoding differences in binary data between Windows/MSSQL and Linux/MySQL
  - Dealing with time zone and date format differences
  - Implementing support for MSSQL's password hashing method at the application level to allow members' existing passwords to be usable in the new system
- Before contracting UAR to work on the project, the client had worked with a developer who spent weeks learning the client's business and creating an estimate for the project, only to bail shortly after beginning work (most likely after realizing their estimate was an order of magnitude too low). UAR worked with the client to rescue the project and break the work into manageable phases.
- Building such a large application on top of WordPress was a fruitful challenge for UAR. Several features and plugins of WordPress required better database indexing and code improvements to be able to handle the size of this client's member base.

NEXT STEPS:

I hope the information provided above is useful and that we can explore working together further. If you have questions, requests, or feedback, we'd be pleased to hear from you at any time.

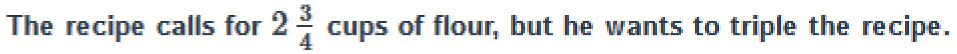
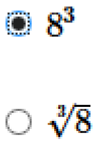
Thank you and Respectfully,

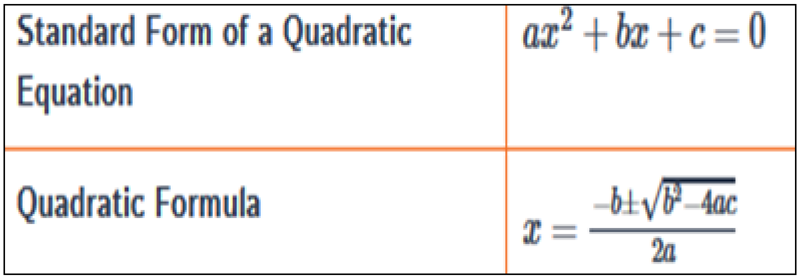
Peter Hanson

Peter Hanson, Founder & CTO
Up and Running Software, Inc.
Cell: 906-281-1178
Email: peter@upandrunningsoftware.com

CASE STUDY 2:

For a client in the education industry, Up and Running Software (UAR) built a testing platform that allows students to take practice tests and prepare for a standard industry test. Schools, adult-education facilities, and correction facilities in the US use this software to help improve the lives of their students. UAR is proud to serve this company as they make tailored content to help adults without much formal education earn high-school equivalency certificates (GEDs and the like).

- The core application is built on the Laravel PHP framework and uses a MySQL database backend. Laravel was selected because it provides a simple, well-designed framework for MVC applications and many supporting tools for common tasks like database migration, unit testing, and command line scripting.
- The user interface is built on Laravel's Blade template system and uses the Twitter Bootstrap framework to standardize CSS styles across the application and across browsers. Bootstrap's grid system is also used to implement a responsive design, thereby supporting a variety of different device screen sizes.
- The application's dynamic behavior is built in JavaScript on top of the jQuery library. jQuery provides a consistent cross-browser JavaScript API and built-in functions that accomplish many common DOM-manipulation tasks.
- In addition to standard multiple choice type questions, UAR also built support for more advanced question types, including:
    - Questions answered by dragging and dropping images to specific positions
    - Questions requiring the student to mark specific points on a graph
    - Short answer questions that accept mathematical expressions and evaluate them to determine correctness
    - Essay-type questions requiring manual grading
- The rich front-end behaviors for these advanced question types were built using client-side JavaScript, allowing for responsive and highly-intuitive interaction across a wide range of devices.
- Using the MathJAX library, UAR implemented support for MathML based formulas on the user interface. These formulas are used on math tests to better present questions and answer choices. Examples of this functionality include:
    - Rendering fractions with question prompts:
        - 
          The recipe calls for $2\frac{3}{4}$ cups of flour, but he wants to triple the recipe.
    - Rendering exponents and root operators in question choices:
        - 
          ● 8^3
          ○ $\sqrt[3]{8}$
    - Providing the student with a table of standard formulas



Standard Form of a Quadratic Equation	$ax^2+bx+c=0$
Quadratic Formula	$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$

- The application supports integration with third-party authentication systems through use of the SAML protocol and has been integrated with CornerStone OnDemand.
- One of the client's legacy applications was integrated to support shared authentication between the old and new applications. This allows users to log into either application using the same credentials and provides the ability for a user to transfer their active session from the old application to the new application without needing to re-authenticate.
- To improve the overall quality of the application, UAR built a tool for synchronizing database content changes between environments. The tool allows content changes to be exported from a database, stored in a version control system, and then loaded into other environments. The design allows any database record to be either locally unique to one environment or shared between multiple remote environments. The tool also supports exporting and importing relationships between database records through the use of GUIDs. This tool allows the client to maintain separate production, staging, and development environments and use those environments to improve their quality-control processes.
- Some customers of the client are unable to connect their facilities to the Internet, so an offline version of the application was developed to run on isolated servers on local networks at those facilities. The offline version includes a C#-based installer that installs and configures a web server, database server, and the testing platform application code.

NEXT STEPS:

I hope the information provided above is useful and that we can explore working together further. If you have questions, requests, or feedback, we'd be pleased to hear from you at any time.

Thank you and Respectfully,

Peter Hanson

Peter Hanson, Founder & CTO
Up and Running Software, Inc.
Cell: 906-281-1178
Email: peter@upandrunningsoftware.com

CASE STUDY 3:

Our client is a leading provider of software that manages used equipment return, inspection, resale and tracking. The company works with some of the largest providers of computing and heavy equipment in the world to handle end of lease return of assets. The application is built to handle a variety of user roles, requirements and customer entities; be it the lessee, lessor, resale distribution or the various members of the client management team.

The application is implemented using PHP with MySQL as the database layer. Smarty is used for the templating system with extensive use of AJAX to improve the end-user experience.

Up and Running is fully integrated into the client's development and functional verification workflow based on the Agile/Scrum methodology. Our team is responsible for implementing end-customer and client feature requests while troubleshooting customer bug reports. Up and Running QA resources handle functional and verification testing while our project manager handles the role of ScrumMaster for the project.

So far, Up and Running has implemented the following aspects of the application's development:

- Improving development efficiency by implementing Agile/Scrum techniques and workflows
- Implementation of various aspects of a customer-facing return management system
- Addition of new workflows and steps to support the needs of new and targeted customers
- Improving reporting and tracking features for the client's administration team
- Extensive refactoring of the application to use more modern PHP methods such as PDO to avoid security pitfalls and improve maintainability while also supporting more recent PHP versions
- Design and implementation of functional and regression testing protocols to ensure application correctness and improve development throughput

Going forward, the client and Up and Running will be:

- Expanding the application to support new sales verticals which require alternate asset and user workflows
- Completing the customer-facing return management system
- Processing feature requests in the task queue as requested by customers and the client/Scrum Product owner
- Continuing refactoring of the codebase to improve extensibility and maintainability

It has been a very interesting project in which to participate. The domain knowledge required for the application is extensive and it has helped us to improve techniques for maintaining and distributing this information in order to quickly bring new team members up to speed. Outside of our normal development contributions, Up and Running has provided value to the client by recommending and implementing a new development workflow based on Agile, which has improved development throughput significantly while also greatly improving task queue visibility for upper management.

NEXT STEPS:

I hope the information provided above is useful and that we can explore working together further. If you have questions, requests, or feedback, we'd be pleased to hear from you at any time.

Thank you and Respectfully,

Peter Hanson

Peter Hanson, Founder & CTO
Up and Running Software, Inc.
Cell: 906-281-1178
Email: peter@upandrunningsoftware.com

CASE STUDY 4:

Our client is a leading developer of software in the healthcare and patient management market. The application with which Up and Running is currently working is built to manage the entire lifecycle of patient medication management. This includes tracking patient home medications, through doctors' orders for additional medications and finally to the documentation that is sent home with the patient upon discharge from the medical facility.

The application is highly customizable for each environment in which it is deployed. In addition, all interactions with the system are audited and logged for full government compliance. On the back-end, the application integrates seamlessly with the hospital network's IT infrastructure through the use of HL7 and other custom XML integration protocols.

The application is implemented using PHP with Zend framework and PostgreSQL as the database with Memcache for caching. For the display layer the application uses Smarty as its templating system with extensive use of AJAX to streamline the physician and nurse workflow.

Up and Running development and QA staff are an integral part of the client's development group for this application. The development workflow is a modified Agile model with story points, sprints and code review.

Our team has participated in the development of the following integral features of the application:

- Enhanced formulary management interface to allow for easier mapping between the full list of possible medications and those that the hospital network maintains internally
- Health Level Seven (HL7) Clinical Document Architecture (CDA) implementation and certification such that the CDA XML document format can be used as an interchange format with other healthcare infrastructure
- Added extensible event architecture to perform certain tasks upon specific and customizable physician and nurse actions
- Implemented numerous customer and client requested features for better data visibility, migration and usability
- Troubleshot various facility-specific environment issues that customers have reported
- Extensive refactoring and reorganization of the system codebase which has evolved over a decade

The client and Up and Running are currently focused on the following roadmap items:

- Extensive expansion to fully support e-prescribing protocols and requirements for controlled substances; the end goal is to replace the doctor's prescription pad entirely
- Updates to support the requirements for facilities and certifications for hospital networks outside of the United States
- Improving support for ambulatory workflows to better support the needs of certain hospital networks and configurations

- Implementation of integration and API services for use by 3rd party vendors
- Continued evolution and refactoring of the codebase to improve maintainability

Our relationship with this client has been quite mutually beneficial. We have provided the client with full development workflow assistance from prototyping through to QA. The client, in turn, has provided our development members with extensive training and knowledge that is specific to the world of the US healthcare market and associated requirements. It has been an interesting project in which to participate and the value to the customer healthcare networks is tangible and very rewarding to the team.

NEXT STEPS:

I hope the information provided above is useful and that we can explore working together further. If you have questions, requests, or feedback, we'd be pleased to hear from you at any time.

Thank you and Respectfully,

Peter Hanson

Peter Hanson, Founder & CTO
Up and Running Software, Inc.
Cell: 906-281-1178
Email: peter@upandrunningsoftware.com

JavaScript CASE STUDIES:

- We did a lot of work over the last year for a venture-funded startup focused on adaptive commerce with the goal of personalizing shopping beyond what standardly exists today. Their project was an e-commerce overlay system where they augmented existing catalog data with personalized filters and recommendations for the data sets of their client websites. It handled injected JavaScript, had Cross-origin resource sharing (CORS) compatibility, had NoSQL data stores (Mongo and ElasticSearch), and used lightweight custom JavaScript objects in their newer implementation. More information regarding some of the items we helped with:
    - Worked on a tool that could be injected into a customer website to determine if there were any glaring issues with how the toolset would run in that environment.
    - Helped analyze and document how their primary configuration generation tool operated.
    - Architected and organized how their new version of their platform would ultimately be used to manage a new site implementation from start to completion.
    - Debugged a series of performance issues with an older deployment of their client-side tools.
- For one client, who offers omnichannel personalization for clients like Best Buy, CDW, Target, Wine.com, Neiman Marcus, Costco, Burton, and Walmart, we helped write up a set of tools and tests to help ensure stability of the client-side libraries that their customers use. More information about how we contributed:
    - Built out a series of comprehensive behavior and unit tests for their client-side injectable component (custom JavaScript objects), which were executed through Jasmine. The component provided all of the access into their core system. These tests ensured that stability would be preserved when the component was modified (it would trip a test either from a function signature issue or behavior change). If there was an error or bug in the component, it would break all integrations with the backend solution so it was an important set of tests. There were approximately 1,500 assertions that went into the test suite. QUnit is another viable alternative that we explored for test execution.
    - Built out an injectable detection tool that would execute a series of validation checks to determine if the customer had their local environment properly configured and was using the injectable component correctly.
    - Designed the detection tool to accept and deal with CORS issues.
    - Set up the detection tool to be library light so a minimalistic set of components were required to make the tests/validation work.
    - Created the tests so they could be used in a harness where the page could be loaded and the tools injected and the test results parsed so the harness could be run on a schedule to keep an eye on customer sites to ensure that they continued to stay compatible with the client-side tool.
- We also did a lot of work for a Fortune 10 customer over the last couple of years. Their system dealt with nuclear power plant inventory and procurement, and it utilized both AngularJS and

Twitter Bootstrap in its construction. It used a typical Java enterprise stack (jBoss, Java, and Oracle) for the service layer. Most of the functionality was handled client-side in AngularJS however. More information about the project:

  - AngularJS helped as it allowed good separation of concerns to be enforced on the client side. We were able to model the data for users, plants, inventory, parts, etc., all on the client side. This information would be saved and managed there and synchronized with the service processes. It resulted in a streamlined experience for the user that was exceptionally fast, one of their core requirements. Along with that, the breakdown of services and controllers and ensuring that DOM manipulation stayed in directives allowed easy testing of the application's code.
  - Twitter Bootstrap was also used to normalize a lot of presentation items in the application. Since it had to support a wide array of browsers (IE8 support included), as much as possible was used from the frameworks to ensure consistency.
  - A feature requirement was mobile support so the groundwork was laid for responsive design through Bootstrap’s use of device-specific classes.
  - D3 was used for a series of reports and pie-chart graphing in the core UI, showing break-out classifications for various parts.
  - We were called on to help the client’s internal systems team troubleshoot environments and issues unrelated to the work that was performed (the code worked well in a newly-built environment). We used our systems experience and troubleshooting skills to identify issues with application and database configurations from a standpoint of what REST endpoints were doing in the application and how the service layer was handling those requests. Since access to infrastructure was restricted, the solution-discovery approach was a matter of reconstructing test cases and coding out a race condition situation to demonstrate how session data was getting stored inconsistently from one instance to the next in their jBoss cluster. We also assisted with the execution of patch scripts and debugging when different environments went out of sync with the datasets they supported.

- A startup selected us to be their interim CTO in 2014 because of how well that project went. It was a very enjoyable project because we were involved from the start to where they are now. Things have been going so well that the founder was able to quit his day job due to the success of the business he created and bootstrapped. The toolset consists of a widget toolset that can be injected into third-party websites by web administrators. It handles API calls to a centralized service to serve up survey configurations and capture form data submissions that get posted back. The widget has a lot of flexibility in its behavior and how it renders for a given site. The application build out was approached in a Minimum Viable Product (MVP) manner, allowing for the main application to rapidly be prototyped and made public for paying-customer use. Many of the latest features have evolved from specific requests by existing customers' usage, which was the intent of the approach. More information:
  - JavaScript-specific object structure for rendering the widget and providing a data model for the widget to utilize on the client side.

  - Use of closures for code encapsulation.
  - A basic initialization signature was used so the project could use a version of jQuery that was already available on the site. jQuery is the only dependency of the toolset.
  - CORS configuration in place to allow the cross-domain API calls from the JavaScript classes.
  - Flexible client-side data embedding that can be sent along with form submissions. This enables detailed data analysis of different demographic lines for specific websites.
  - Implementations of different form factors of sites (desktop versus mobile).
  - Use of ElasticSearch on the backend for survey storage and form post handling.
- For one client, we used websockets to connect a number of browsers together to control the presentation and answering of survey material. A controlling interface would show questions and results and use websockets to communicate state changes, which would route though the app system running on Node.js, and in turn push out the notifications to the other clients. Clients could register/remove themselves from the app and subscribe to a certain channel. The system also handles if a client drops off and comes back online, tracking the socket ID signatures for different clients and tracking their connection status.

NEXT STEPS:

We hope the information provided above is useful and that we can explore working together further. If you have questions, requests, or feedback, we'd be pleased to hear from you at any time.

Thank you and Respectfully,

Peter Hanson

Peter Hanson, Founder & CTO
Up and Running Software, Inc.
Cell: 906-281-1178
Email: peter@upandrunningsoftware.com