

Objective

Build high value software in a smart, passionate, caring team

As a software engineer I value UX design, simplicity (as in <https://www.infoq.com/presentations/Simple-Made-Easy>), maintainability, absence of fear (static typing, automatic tests) and working as a team.

As a team member/team lead I value shared vision, aligned goals, care and empathy towards mission, results and people.

Skills

Programming Languages

- JavaScript (know good and bad parts, prefer Ramda to underscore)
 - TypeScript (+DefinitelyTyped, I generally prefer static to dynamic typing))
 - Flow (+flowtyped)
- C# (co- and contra- variance, oldie-but-goodie <https://vimeo.com/44287613>)
- Python 2.7
- Java
- SQL (T- and PL/ dialects)
- ReasonML/OCaml (know what polymorphic variants and type constraints are)
- F#
- Elm (version that had signals)
- Haskell (can read basic code samples)
- C/C++11 (I used to know them a long time ago)

Concepts

- Client side architectures (Flux, Redux, unidirectional data flow, MVVM etc)
- Tactical DDD (no strategic DDD and CQRS)
- OOD/OOP, SOLID
- FP (`Ma -> (a->Mb) -> Mb`)
- ORM
- Fast (Unit) and slow (integration, acceptance) testing
- Web APIs (RESTful and not)

Platforms, Frameworks and Technologies

- **JavaScript:** Of probable interest: Angular, React, Knockout.js, Jest, Jasmine, jQuery, underscore.js,
- **.NET:** Castle Windsor, NHibernate, Entity Framework, ADO.NET, ASP.NET MVC, ASP.NET WebApi, WCF, LINQ, TPL, NUnit, Moq
- **Nodejs:** Express
- **Databases:** Firebase, MongoDB, BigQuery, SQL Server 2005/2008/2012, Oracle 11g, T-SQL, PL/SQL, SQL Server Reporting Services.
- **VCS:** Git, Mercurial
- **Operations:** Kubernetes+Helm, GCP - GKE, AppEngine, Prometheus
- **Cloud:** Google BigQuery, Google ..., AWS SQS, AWS S3
- **SDLC tools:** Travis, CircleCI, Jira, TeamCity, Jenkins
- **Virtualization:** Docker

Employment

2016 - August 2018, Senior Engineer, Pear Deck, Iowa City, IA

We developed a web tool to help teachers engage students, especially in the challenging environments with large class sizes so everybody could be heard and be safe. We consistently pushed out new features while keeping system up and running and bringing in new developers. Our team did the full cycle of product development - from UX design to implementation to ongoing support.

We were able to do that by working in small vertical teams, constantly aligning, and having a shared vision on what product, code and process should be. We never left anybody alone with problem, situation or task. As one of two most-experienced devs, I put steady efforts into gradually improving product architecture, code quality, system maintainability, and most importantly creating a shared vision on where we are going technically as a development team.

Implemented collection of microservices running on GKE/AppEngine using JavaScript, Node, Java, Python. Implemented front-ends in Javascript, Angular, Elm, Reason ML+React.

Provided product/UX design ideas, suggestions and feedback.

Tested automatically (Jest, Selenium, unit, integration and acceptance tests) and manually.

Made new developers productive and helped them with doing right things for company, team and themselves. Reviewed PRs, provided help and suggestions.

Wrote deployment procedures, k8s configurations (including automatic load balancing), configured GCP. Monitored application up-time and performance, detected and fixed application and infrastructure problems.

2012 - 2016, Senior Application Developer, University of Iowa, Iowa City, IA

Worked on university wide procurement and travel expense management system used by University Of Iowa and University of Northern Iowa.

Focused on most painful problems and problems that no one else was well equipped to work on.

Worked on building a professional software development team, set up organization processes, established shared vision, created and maintained standard operating procedures (SOP), and supervised other developers.

Talked to customers, understood their problems, proposed and discussed solutions. Wrote specs, created UI wireframes, and project estimates. Generally, made things go forward.

Wrote front-ends in Angular, Knockout, jQuery, Telerik web controls using Bootstrap, AngularStrap (the mess of technologies is the result of having to support many systems, including very old ones).

Wrote DDD-ish back-ends (domain model, repos, services, aggregate roots, domain events) in C#, LINQ for SQL, SQL, EF going against SQL Server and Oracle. All components wired together using IoC container (Windsor). From no coverage to full coverage by fast (unit) and slow (integration) tests.

Wrote installation (Desired State Configuration) and deployment (we had our own Octopus-like deployment service) scripts, created a configuration management DSL that was used to describe queryable infrastructure model. Automated existing processes, created utilities for Active Directory, Confluence, TeamCity, Sql Server and Oracle.

Implemented shared library of components to codify best practices (for front-end as well as for back-end). Made efforts to introduce policies regarding code sharing and continuous integration/deployment.

Persisted in establishing team protocols (e.g. "Ask for help", "Check in", "Double-check", "Shared vision").

Wrote SSRS reports, wrote a rdl report formatter that allowed to define SSRS object styles using selectors and definitions (stylesheets for RDL), and made data analysis tools using F#.

Wrote acceptance tests using home-grown WebDriver based framework that allowed separation of logical and physical concepts. Wrote test plans for manual testing and did that testing myself.

Made presentations and organized seminars to teach colleagues about modern tools and methods. Set up student and staff code reviews and actively provided feedback and challenges for students working for the department.

Installed and maintained TeamCity, Jenkins and Rhodecode installations, administered Windows production and test servers.

1995 - 2012, Head of Engineering, MetaCommunications Inc., Iowa City, IA

Whatever was necessary. Did a lot of stuff including product design and implementation (C++, C#, SQL, a lot of JavaScript), product implementation on customer sites, marketing, testing (automated and manual), office and personnel management.

References

- Valerij Petrulevich, University of Iowa Financial And Business Information Systems, vpetrulevich@uiowa.edu
- Joel Jeddelloh, Senior Software Engineer, joel@peardeck.com
- Dan Sweeney, VP of Product, dans@peardeck.com