

# TWIN CITIES SOFTWARE SYMPOSIUM

OCTOBER 20 - 21, 2023 - MINNEAPOLIS, MN

2023 NO FLUFF JUST STUFF TOUR

WORLD-CLASS TRAINING FOR SOFTWARE DEVELOPERS & ARCHITECTS

Software Architecture · Modern Java · Cloud · Docker · Microservices · Kubernetes · Kotlin · Angular · React · Vue.js · JavaScript · Machine Learning · Functional Programming · Web App Security · Spring · Testing

## Attend In-Person or Online

We are excited to come together again for in-person events! However, you will have the option to join us virtually via Zoom.

In addition, attendees will have access to video recordings of all sessions.

## Why Attend the NFJS Tour?

No Fluff Just Stuff is focused on delivering an educational experience, free from vendor bias. NFJS features the best speakers in the industry with knowledge and passion for teaching. Our iterative content is updated not only year-to-year but, week-to-week!

### Learn from the Best

Our speakers are not vendor representatives. They are industry recognized experts. They are published authors, consultants, executives, and open source leaders.

### In-Depth 90-Minute Sessions

Our longer session format, workshops, and multi-part sessions allow speakers to go in-depth and teach the detailed concepts you need to know.

### Agile Practices

Our speakers emphasize and present on topics such as: Test Driven Development, Continuous Integration, Code Quality Measurements, Code Smells, Team Building, and Customer Collaboration.

### Understand Web Security

The web is an increasingly hostile environment for web applications. The NFJS Tour includes security focused sessions and workshops so you will understand best security practices.

### Develop Your Soft Skills

Effective engineers need significant technical depth and breadth and domain knowledge. In addition, there is another perhaps more vital aspect of being an architect - the soft skills. Communication, leadership, persuasion, and more.

### Cloud Architectures

The NFJS Tour explores different cloud computing architectures and how you can take advantage of them.

<https://nofluffjuststuff.com>

# Twin Cities Software Symposium

## -Session Schedule-

(event schedule as of May 17, 2024)

### Friday, Oct. 20

7:30 - 8:15 AM : REGISTRATION/BREAKFAST - SALON A

8:15 - 8:30 AM : WELCOME - SALON A

8:30 - 10:00 AM - Sessions

#### **Session #1 @ SALON B : Know your Java? by Venkat Subramaniam**

Many of us have significant experience in Java. Yet, from time to time, we get tripped up by some code that we quite did not expect to behave the way it does.

#### **Session #2 @ SALON C : Tailor-Made Software Architecture by Michael Carducci**

Architecture is often described as “the stuff that’s hard to change” or “the important stuff (whatever that is)”. At its core, architecture defines the very essence of software, transcending mere features and functions to encompass vital capabilities such as scalability, evolvability, elasticity, and reliability. But here’s the real question: where do these critical capabilities truly originate? In this session, we’ll embark on a journey to uncover the secrets behind successful architectures. While popular architecture patterns may offer a starting point, it’s time to unveil the startling truth; both monolith and microservices-based projects continue to stumble and falter at alarming rates. The key to unparalleled success lies in the art of fine-tuning and tailor-making architectures to precisely fit the unique needs of your organization, environment, and the teams delivering the software.

#### **Session #3 @ JEFFERSON : What’s New in Spring 6 and Spring Boot 3 by Craig Walls**

In this example-driven session, we’ll review several tips and tricks to make the most out of your Spring development experience. You’ll see how to apply the best features of Spring and Spring Boot, including the latest and greatest features of Spring Framework 6 and Spring Boot 3.

#### **Session #4 @ WASHINGTON : Kafka Fundamentals by Daniel Hinojosa**

Kafka is a “must know”. It is the data backplane of the modern microservice architecture. It’s now being used as the first persistence layer of microservices and for most data aggregation jobs. As such, Kafka has become an essential product in the microservice and big data world.

10:00 - 10:30 AM : BREAK: BALLROOM FOYER

10:30 - 12:00 PM - Sessions

#### **Session #5 @ SALON B : Structured Concurrency in Java by Venkat Subramaniam**

A prerelease feature in Java, structure concurrency is going to change how we do concurrent programming.

#### **Session #6 @ SALON C : An Architect’s Approach to API Strategies by Michael Carducci**

Integration, once a luxury, is now a necessity. Doing this well, however, continues to be elusive. Early attempts to build better distributed systems such as DCOM, CORBA, and SOAP were widely regarded as failures. Today the focus is on REST, RPC, and graphql style APIs. Which is best? The go-to answer for architects is, of course, “it depends.”

#### **Session #7 @ JEFFERSON : Spring into K8s: Deploying Spring Application in Kubernetes by Craig Walls**

Although Java originally promised write once, run anywhere, it failed to fully deliver on that promise. As developers, we can develop, test, and build our applications into WAR or executable JAR files and then toss them over the wall to a Java application server and Java runtime that we have no control over, giving us zero confidence that the application will behave the same as when we tested it. Containers fulfill the write-once, run anywhere promise that Java wasn’t able to, by packaging the runtime and even the operating system along with our application, giving greater control and confidence that the application will function the same anywhere it is run. Additionally, containers afford several other benefits, including easy scaling, efficiency in terms of resource utilization, and security by isolating containers from their host system and from other containers. While deploying Spring applications in containers has always been possible, Spring Boot makes it easier to containerize our applications and run them in container architectures such as Kubernetes. Spring Boot’s support for containerization includes two options: Creating containers based on buildpacks or using layers as a means of modularizing and reducing the size of our application deployments. Moreover, new components in the Spring ecosystem can make your Spring applications Kubernetes-savvy so that they can take advantage of what a containerized architecture has to offer.

#### **Session #8 @ WASHINGTON : Kafka Fundamentals (continued) by Daniel Hinojosa**

Kafka is a “must know”. It is the data backplane of the modern microservice architecture. It’s now being used as the first persistence layer of microservices and for most data aggregation jobs. As such, Kafka has become an essential product in the microservice and big data world.

12:00 - 1:00 PM : LUNCH: SALON A

# Twin Cities Software Symposium

## -Session Schedule-

(event schedule as of May 17, 2024)

1:00 - 2:30 PM - Sessions

### **Session #9 @ SALON B : Pattern Matching in Java by Venkat Subramaniam**

The switch feature of Java has gone through an amazing transformation. In this presentation will start with switch as a statement, transform from there to an expression, and into a full blown pattern matching syntax.

### **Session #10 @ SALON C : Hypermedia and the rest of REST by Michael Carducci**

REST is, undoubtedly one of the most maligned and misunderstood terms in our industry today. So many different things have been called REST, that the world has virtually lost all meaning. Many systems and applications that self-describe as “RESTful”; usually are not, at least according to REST as defined in Dr. Roy T. Fielding’s 2000 Dissertation, “Architectural Styles and the Design of Network-based Software Architectures”. The wild success of the architecture derived by Dr. Fielding led many to want to emulate it (even when it was inappropriate to do so). As a shorthand, organizations began referring to “RESTful” systems, which exposed “RESTful” APIs. Over time “REST” became a buzzword referring to a vague generalization of HTTP/json APIs that typically bear little to no resemblance to the central ideas of REST (and thus elicit few of the benefits). Hypermedia is the central pillar and defining characteristic of the REST architectural style yet it remains almost universally absent.

### **Session #11 @ JEFFERSON : Reactive Spring APIs by Craig Walls**

In this example-driven presentation, we’ll focus on how to build reactive APIs in Spring. We’ll start with Spring WebFlux, a reactive reimagining of the popular Spring MVC framework for HTTP-based APIs. Then we’ll have a look at RSocket, an intriguing new communication protocol that is reactive by design.

### **Session #12 @ WASHINGTON : Measuring your architecture by Raju Gandhi**

It’s not just architecture’s evolutionary architecture. But to evolve your architecture, you need to measure it. And how does that work exactly? How does one measure something as abstract as architecture? In this session we’ll discuss various strategies for measuring your architecture. We’ll see how you know if your software architecture is working for you, and how to know which metrics to keep an eye on. We’ll also see the benefits of measuring your architecture.

2:30 - 2:45 PM : BREAK: BALLROOM FOYER

2:45 - 4:15 PM - Sessions

### **Session #13 @ SALON B : Designing Microservices: From Architecting to Data Modeling by Venkat Subramaniam**

How does designing Microservices differ from designing more traditional applications? What is a better way to learn than to take a problem, analyze the requirements, explore the design options, apply the concepts of bounded context, and arrive at the architecture and design of Microservices to realize the requirements?

### **Session #14 @ SALON C : The Linked Data Revolution is here, are you ready? by Michael Carducci**

The web is arguably the single most impactful revolution in human history (to date). By agreeing on a simple set of standards, we have collectively unlocked all the world’s information. Documents can be discovered, retrieved, published, and shared so easily we don’t even think about it. Data, on the other hand, is a different story. Our data remains stuck in the 1980s. Locked in silos, each with a different format, interface, and conventions that must be interpreted by a human, parsed, mapped, and converted. Data is at the heart of many problems we solve today, and we produce data exponentially faster than we can consume it.

### **Session #15 @ JEFFERSON : Reactive Spring Persistence by Craig Walls**

In this example-driven presentation, we’ll focus on working with reactive data persistence. We’ll start by seeing how to create reactive repositories for relational databases with Spring Data R2DBC. Then we’ll explore non-relational reactive persistence for MongoDB and Cassandra.

### **Session #16 @ WASHINGTON : Advanced Git by Raju Gandhi**

You have been using Git for a while. You know how to stage and commit your work, create and delete branches and collaborate with your team members using remotes. But Git often leaves you confused “ever committed to your work to the wrong branch? Even worse, ever accidentally delete a branch that you needed to keep around? And what is God’s good name is “Detached HEAD state”? Why tag commits, when we have branches? Is there a better work-flow than just using merges? What’s the difference between a merge and a rebase? The answer to all of these questions, and more, lies in the constitution of a commit, and the directed acyclic graph (DAG) that Git uses to manage your history. This, right here, is the key to understanding everything in Git.

4:15 - 4:30 PM : BREAK: BALLROOM FOYER

4:30 - 6:00 PM - Sessions

# Twin Cities Software Symposium

## -Session Schedule-

(event schedule as of May 17, 2024)

### **Session #17 @ SALON B : Designing Microservices: From Architecting to Data Modeling (continued) by Venkat Subramaniam**

How does designing Microservices differ from designing more traditional applications? What is a better way to learn than to take a problem, analyze the requirements, explore the design options, apply the concepts of bounded context, and arrive at the architecture and design of Microservices to realize the requirements?

### **Session #18 @ SALON C : The Linked Data Revolution II - Tools, patterns, and practices by Michael Carducci**

Part one of this series introduces the ideas, motivations, and applications of linked data along with historical context. This more technical session dives deeper into the tech stack and available tooling.

### **Session #19 @ JEFFERSON : Spring Graph QL by Craig Walls**

In this example-driven session, we're going to look at how to implement GraphQL in Spring. You'll learn how Spring for GraphQL builds upon GraphQL Java, recognize the use-cases that are best suited for GraphQL, and how to build a GraphQL API in Spring.

### **Session #20 @ WASHINGTON : Advanced Git (continued) by Raju Gandhi**

You have been using Git for a while. You know how to stage and commit your work, create and delete branches and collaborate with your team members using remotes. But Git often leaves you confused &#8212; ever committed to your work to the wrong branch? Even worse, ever accidentally delete a branch that you needed to keep around? And what is God's good name is &quot;Detached HEAD state&quot;? Why tag commits, when we have branches? Is there a better work-flow than just using merges? What's the difference between a merge and a rebase? The answer to all of these questions, and more, lies in the constitution of a commit, and the directed acyclic graph (DAG) that Git uses to manage your history. This, right here, is the key to understanding everything in Git.

6:00 - 6:45 PM : DINNER: SALON A

6:45 - 7:30 PM : Keynote: 30 Years of Software Development: Lessons from the Trenches - Kirk Knoernschild

## **Saturday, Oct. 21**

7:30 - 8:15 AM : BREAKFAST: SALON A

8:15 - 9:45 AM - Sessions

### **Session #21 @ SALON B : Asynchronous Programming in Java using Virtual Threads by Venkat Subramaniam**

Threads have been part of Java since the beginning. But, the new virtual threads, introduced as prerelease in Java 19, are different in how they're implemented and how we can benefit from them.

### **Session #22 @ SALON C : Personal Knowledge Management - Second Brain Methods and Madness by Michael Carducci**

We are knowledge workers and ultimately, we must own our growth and learning. [Personal Knowledge Management]([https://en.wikipedia.org/wiki/Personal\\_knowledge\\_management](https://en.wikipedia.org/wiki/Personal_knowledge_management)) is a process of collecting information that one uses to gather, classify, store, search, retrieve and share knowledge in their daily activities and the way in which these processes support work activities. Despite taking notes, bookmarking web content, and highlighting passages in books; often we struggle to recall or rediscover these many insights we pick up daily in our work and life. This session introduces a tool and some process recommendations to never again lose discoveries and knowledge resources.

### **Session #23 @ JEFFERSON : Agile Architecture by Kirk Knoernschild**

In today's volatile technology and business climate, big architecture up front is not sustainable. Attempts to define the architectural vision for a system early in the development lifecycle does not work. To accept change, teams are moving to agile methods, but agile methods provide little architectural guidance. In this session, we provide practical guidance for software architecture on agile projects.

### **Session #24 @ WASHINGTON : Machine Learning Data Pipelines by Daniel Hinojosa**

This workshop builds an entire event driven data pipeline with Machine Learning and Kafka. From Kafka where we use producers or Kafka Connect to generate information, we then will Kafka Streams to apply a machine learning model to make business decisions.

9:45 - 10:00 AM : BREAK: BALLROOM FOYER

# Twin Cities Software Symposium

## -Session Schedule-

(event schedule as of May 17, 2024)

10:00 - 11:30 AM - Sessions

### **Session #25 @ SALON B : Non-Language Changes in Java by Venkat Subramaniam**

Java has been evolving at a rapid pace. Some of the most noticed changes are the features in the language. However, there are other interesting and significant changes in Java, not related to the language but in the JDK and the ecosystem.

### **Session #26 @ SALON C : The Linux Shell - From Novice to Wizard by Michael Carducci**

The linux command line is powerful, but for those of us who weren't borne into that world, it can be a bit daunting. If the extent of your usage of the command line is limited to a handful of memorized git commands, npm cli, and the occasional copy and paste one-liner from the web, this session is for you.

### **Session #27 @ JEFFERSON : Refactoring Monolithic Software by Kirk Knoernschild**

Monoliths are out and microservices are in. Not so fast. Many of the benefits attributed uniquely to microservices are actually a byproduct of other architectural paradigms with modularity at their core.

### **Session #28 @ WASHINGTON : Machine Learning Data Pipelines (continued) by Daniel Hinojosa**

This workshop builds an entire event driven data pipeline with Machine Learning and Kafka. From Kafka where we use producers or Kafka Connect to generate information, we then will Kafka Streams to apply a machine learning model to make business decisions.

11:30 - 12:15 PM : EXPERT PANEL DISCUSSION: SALON A

12:15 - 1:00 PM : LUNCH: SALON A

1:00 - 2:30 PM - Sessions

### **Session #29 @ SALON B : Refactoring Code: An Incremental and Purpose Driven Approach by Venkat Subramaniam**

Continuous refactoring is critical to succeeding in projects and is an important part of sustainable agile development.

### **Session #30 @ SALON C : Mob Mentality - Collaborative coding and problem solving by Michael Carducci**

Mob Programming is a style of programming in which the entire team sits together and works on a single task at a time. Teams that have worked this way have found that many of the problems that plague normal development just melted away, possibly because communication and learning increases. Teams also find that the quality of their code increases. They find their capacity to create increases. However, the best part of all this is that teams end up being happier and more cohesive.

### **Session #31 @ JEFFERSON : Understanding the Java Platform Module System by Kirk Knoernschild**

As of Java 9, modularity is built in to the Java platform...Finally! Yet few teams are using it. And in reality, you may never use it...at least not for a while. However by understanding the module system, you're guaranteed to see the Java platform in a completely different light. In this session, we explore the default module system, how it works on the Java platform, and what's in the future for the Java Platform Module System.

### **Session #32 @ WASHINGTON : DDD and Microservices by Raju Gandhi**

We live in a world of microservices. Yet, what is a microservice? What defines the boundaries of a microservice? How do we define the relationships between microservices? Thankfully domain-driven design gives us the concepts and practices to better design and decompose our services. In this session we will consider many of the concepts of DDD: How bounded contexts use Ubiquitous language to model the domain, how context maps can be used to establish the interconnections between services as well aggregates and domains events, all of which will service us well as we go about creating our microservices. We will also discuss the "tactical" patterns of DDD. We will see how we can "embed" the ubiquitous language in code, and the architectural influences of DDD.

2:30 - 2:45 PM : BREAK: BALLROOM FOYER

2:45 - 4:15 PM - Sessions

### **Session #33 @ SALON B : Refactoring Code: An Incremental and Purpose Driven Approach (continued) by Venkat Subramaniam**

Continuous refactoring is critical to succeeding in projects and is an important part of sustainable agile development.

### **Session #34 @ SALON C : Mob Mentality - Collaborative coding and problem solving (continued) by Michael Carducci**

Mob Programming is a style of programming in which the entire team sits together and works on a single task at a time. Teams that have worked this way have found that many of the problems that plague normal development just melted away, possibly because

# Twin Cities Software Symposium

## -Session Schedule-

(event schedule as of May 17, 2024)

communication and learning increases. Teams also find that the quality of their code increases. They find their capacity to create increases. However, the best part of all this is that teams end up being happier and more cohesive.

### **Session #35 @ JEFFERSON : Testing Legacy Code & Avoid Mocking by Daniel Hinojosa**

Our jobs usually deal with something other than new code. It is usually old spaghetti and difficult-to-read code. How do we test such code? How do we get through it? How can we surgically remove and make some of this harmful code testable?

### **Session #36 @ WASHINGTON : DDD and Microservices (continued) by Raju Gandhi**

We live in a world of microservices. Yet, what is a microservice? What defines the boundaries of a microservice? How do we define the relationships between microservices? Thankfully domain-driven design gives us the concepts and practices to better design and decompose our services. In this session we will consider many of the concepts of DDD; How bounded contexts use Ubiquitous language to model the domain, how context maps can be used to establish the interconnections between services as well aggregates and domains events, all of which will service us well as we go about creating our microservices. We will also discuss the "tactical" patterns of DDD; We will see how we can "embed" the ubiquitous language in code, and the architectural influences of DDD.

4:15 - 4:30 PM : BREAK: BALLROOM FOYER

4:30 - 6:00 PM - Sessions

### **Session #37 @ SALON B : Let's Have Some Fun with Game Of Life: And Learn to Think Functionally Along the Way by Venkat Subramaniam**

Game of Life is an intriguing game. At first look it looks simple, but as you look closer, it appears to be quite complex. How can we implement this game with different constraints, what are the constraints? Is it possible to use functional programming for this, to honor immutability? You see, it is intriguing.

### **Session #38 @ SALON C : Finding Signal in the Noise: The art of Execution by Michael Carducci**

In tech teams it's a constant firefight. We react. Then we react to the reaction... the cycle continues. In all this noise, in all this chaos, how do we move forward. How do we remain proactive?

### **Session #39 @ JEFFERSON : Next Gen Testing Tools for Java by Daniel Hinojosa**

We have been using JUnit and doing TDD for years, but you can take testing further. In this session, we will discuss some tools you absolutely need for testing your code outside of the regular stack you currently use.

### **Session #40 @ WASHINGTON : On being an effective developer by Raju Gandhi**

As developers we not only operate in different contexts, but also often have these different contexts interplay as part of our work. Each of the tools that we use; version control systems like Git (along with collaborative tools like Github/Gitlab), IDE's like Eclipse/IntelliJ, build systems like Gradle, Ci/Cd tooling like Jenkins, IaC tools like Ansible, the command line; all introduce context. To be effective developers we need to know when to operate in a certain context, combine or tease apart how these contexts interplay. Can you improve your release announcements if format your commit messages consistently? You bet! How should your build tool interact with your version control system? What does naming your files have to do with how you use your IDE?