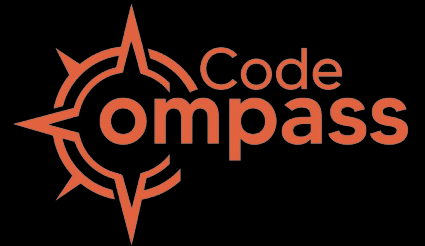# Human Driven Development: Balancing Speed, Quality, and Sanity

—

Joey Pedicini
CodeCompass, LLC

Outline

1. My career journey / lessons learned

2. Impromptu retrospective

3. Mindset - balancing quality and care

4. What is "Human Driven Development"

5. Applying HDD in the workplace

# My Career Journey / Lessons Learned

Lesson 1

Practice being uncomfortable.

1. Saying "Hi" to someone in the hallway in passing

2. Asking a store employee for help finding an item

3. Making small talk with a coworker in the break room

4. Introducing yourself to a new colleague

5. Joining a group conversation at a social event

6. Speaking up during a team meeting

7. Giving a short update or presentation to a small group

8. Networking at a professional event

9. Leading a meeting with multiple stakeholders

10. Giving a formal presentation to a large audience
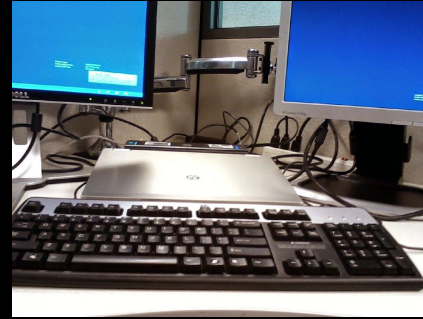
Lesson 2

Try new things. Failure is an opportunity to learn.

1. Writing unit tests for existing code

2. Contributing to an open-source project

3. Pair programming with a senior developer

4. Debugging and fixing a bug in the codebase

5. Implementing a new feature from scratch

6. Refactoring legacy code

7. Learning and using a new programming language

8. Building a side project or personal website

9. Participating in a hackathon

10. Giving a code review to a peer
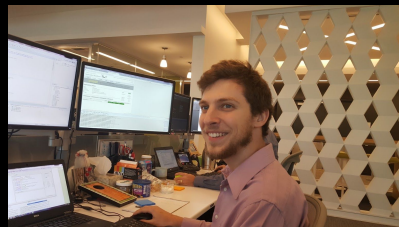
Lesson 3

Be patient, seize the opportunities when they come

1. Working on a high-impact project for a major client

2. Leading a team or project

3. Mentoring junior developers

4. Attending or speaking at industry conferences

5. Contributing to open-source projects

6. Learning and implementing new technologies or frameworks

7. Participating in hackathons or coding competitions

8. Transitioning into a specialized role, such as a data scientist or cybersecurity expert

9. Collaborating with cross-functional teams to deliver a product

10. Taking on a management or leadership role within the organization

Follow your intuition

QA Engineer (2017)

Lesson 4

Prioritize health:
physical, mental, spiritual

1. Take regular breaks using techniques like the Pomodoro method

2. Practice mindfulness or meditation

3. Seek support through therapy or counseling if needed

4. Engage in hobbies and activities outside of work

5. Set boundaries to ensure a healthy work-life balance

6. Incorporate regular exercise into your routine, such as walking, running, or yoga

7. Maintain a balanced diet and stay hydrated

8. Ensure adequate sleep each night

9. Engage in practices that foster inner peace, such as meditation, prayer, or journaling

10. Cultivate gratitude by keeping a gratitude journal or reflecting on positive experiences regularly
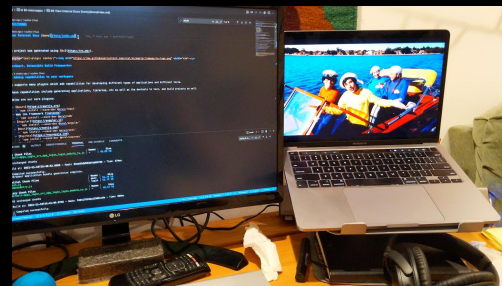
Lesson 5

Find your "why"

1. Family commitments and providing a better future for loved ones

2. Financial gain and achieving long-term financial stability

3. Personal development and self-fulfillment

4. Gaining recognition and respect in your field

5. Pursuing passion and interest in your work

6. Achieving professional goals and milestones

7. Improving quality of life and work-life balance

8. Expanding knowledge and expertise in your area

9. Creating a positive impact and contributing to society

10. Building a legacy and leaving a lasting mark in the industry

If you fall it is because you have risen

Senior Software Engineer - Frontend (2018-2021)

Lesson 6

Focus on your wins

1. Keep a "brag list" of accomplishments and milestones.

2. Maintain a journal to reflect on daily achievements and progress.

3. Celebrate small wins and acknowledge incremental progress.

4. Share successes with peers and mentors for external validation.

5. Set realistic and achievable goals to regularly experience a sense of accomplishment.

6. Ignore the tech hype and focus on personal growth and achievements.

7. Create a portfolio showcasing completed projects and skills.

8. Attend meetups or conferences to present your work and receive recognition.

9. Regularly review and update your resume to see how far you've come.

Lesson 7

Practice the art of creative problem solving

1. Solve puzzles or brain teasers to keep the mind sharp and think outside the box.

2. Take up a new hobby or learn a new skill to stimulate different parts of the brain.

3. Read books or articles outside of your usual interests to gain new perspectives.

4. Participate in improv classes or activities to enhance quick thinking and adaptability.

5. Join a creative writing group to practice storytelling and innovative thinking.

6. Travel to new places to experience different cultures and viewpoints.

7. Collaborate with people from different fields to tackle problems from multiple angles.

8. Play strategy games or board games to develop strategic thinking and foresight.

Focus on what makes you Human

Frontend Architect (2021 - 2023)
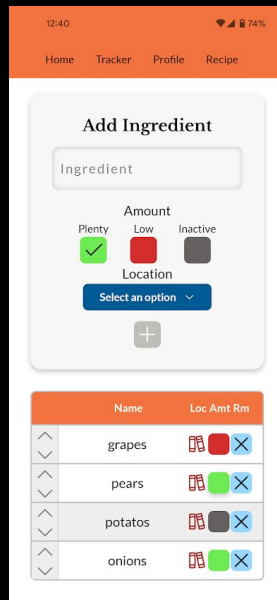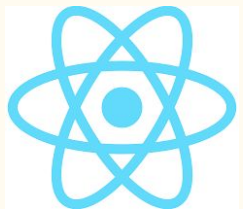
Lesson 8

Master the Art of Communication: Writing and Sharing Ideas

1. Start a blog to write about projects, challenges, and solutions.

2. Contribute to open-source documentation to improve clarity and usability.

3. Write articles for industry publications or websites.

4. Create detailed project documentation for team members and stakeholders.

5. Participate in online forums and communities to share knowledge and answer questions.

6. Develop and share coding tutorials or guides on platforms like GitHub or dev.to

7. Write and present proposals for new projects or initiatives within your organization.

8. Create and share posts on social media platforms.

Lesson 9

Build and Nurture Relationships

1. Connect with professionals on LinkedIn and engage with their content.

2. Attend local meetups, conferences, and industry events to network with peers.

3. Participate in online communities and forums, such as Stack Overflow or Reddit.

4. Collaborate on open-source projects to build relationships with other developers.

5. Share praise and recognition for coworkers' achievements on social media or team channels.

6. Mentor junior developers or offer to help with their projects.

7. Organize or join study groups to learn new technologies and skills together.

8. Schedule regular coffee chats or lunch meetings with industry contacts.

9. Volunteer for events or organizations to meet like-minded professionals.

Lesson 10

Work / Play balance. Have more fun.

1. Set boundaries and clearly define work hours to ensure personal time is protected.

2. Take regular breaks throughout the day.

3. Engage in hobbies and dedicate time to non-work-related activities.

4. Get to know your teammates, set up time to hang out and work together.

5. Pursue side projects or work on personal projects and learn new skills that interest you outside of your main job.

6. Create a workspace that makes you feel comfortable and happy.

7. Gamify tasks by turning work tasks into games or challenges.

8. Celebrate achievements by taking time to celebrate personal and team accomplishments, no matter how small.

Impromptu Retrospective

Why software engineering?

Problem solving
Learning
Discovering

- "It's the thing I'm good at and the thing I enjoy"

- "I love the amount of problem solving and approaches you can take to solve a problem"

- "I get to create my own systems and functionality and play around with new technologies"

- "My learning will never end and I can keep discovering something new and advanced every other day"

What brings you joy in your work?

Problem solving
Accomplishing tasks
Teamwork

- "I like solving problems and solving them well"

- "The feeling of accomplishing a task or ticket"

- "To see the platform evolve"

- "Contributing effectively to organizational goals through teamwork"

Essential Qualities for Success

Problem solving
Communication
Teamwork

- "Leadership, communication, being receptive, and collaboration"

- "Collaboration and teamwork"

- "You have to give a shit"

- "Continuous Learning and Adaptability. Problem-solving ability"

Challenges and Frustrations

Tech Debt
Testing
Planning

- "Adapting to new technologies"

- "How to develop our test system"

- "Being new to the codebase"

- "Running into errors outside of the code I wrote"

- "Tech debt"

- "Unclear Planning"

- "Rituals more important than code"
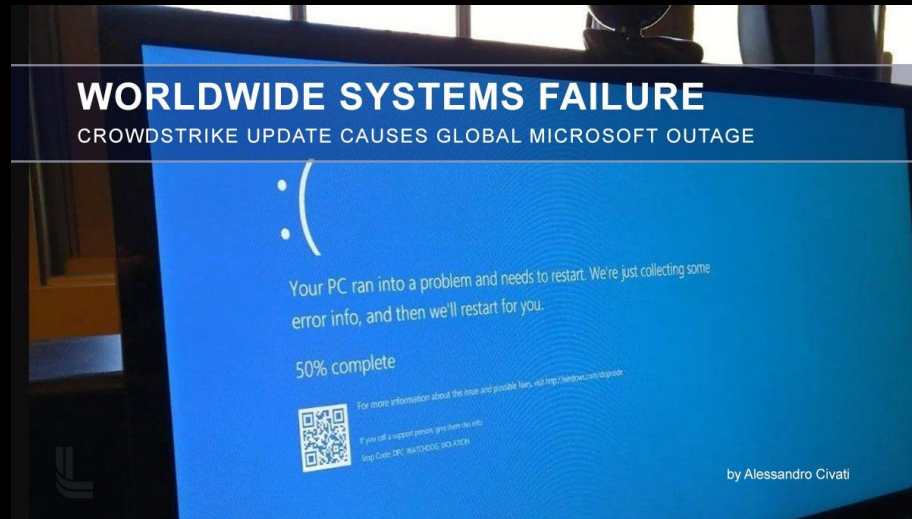
Magic Wand Desired Changes

Tech Debt
Testing
Merge Conflicts

- "A magic debugger that writes automated tests for me"

- "Merge conflicts that fix themselves"

- "A plan on resolving tech debt"

- "Thinking about a11y"

- "Clean up AWS environment and database structure"

Mindset Matters: Balancing Quality and Care

Small mistakes can have big consequences

"CrowdStrike said the outage was caused by an out-of-bounds memory read error beyond the end of the input data array."

WORLDWIDE SYSTEMS FAILURE
CROWDSTRIKE UPDATE CAUSES GLOBAL MICROSOFT OUTAGE

:(

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

50% complete

For more information about this issue and possible fixes, visit http://windows.com/stopcode

If you call a support person, give them this info:
Stop Code: DPC WATCHDOG VIOLATION

by Alessandro Civati

Reduce mistakes by prioritizing how much you "give a shit"

The less you care, the more likely you are to cut corners and make mistakes.

The more you care, the less likely you are to cut corners and make mistakes.

———

Be the change you wish to see

A new software development methodology: Human-driven Development

Joey Pedicini
Human-centered tech | CTO | Nature-inspired dev 🧠🌱
Helping people build products that matter

December 11, 2023

Why don't I enjoy my job?

I get paid well. I work from home. My coworkers are nice. My workload is fairly light. There isn't too much pressure.

What do I not like?

The way projects are run. The relationship between the developers and the managers. The tech stack I am working with. The lack of autonomy. The pointless meetings.

Joseph Pedicini
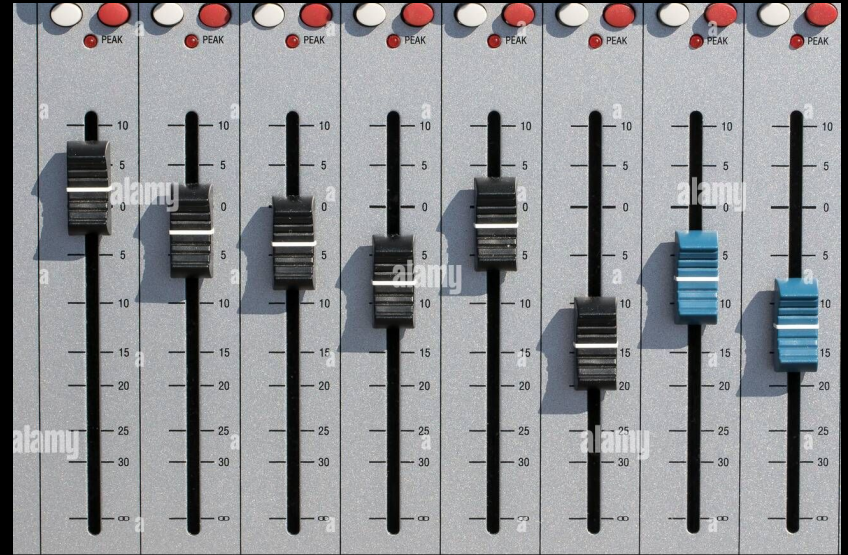This shit drains me #corporate meeting ...
4 months ago

0:00   2:44

You can't care about everything

Performance
Scalability
Readability
Maintainability
Team Velocity
Documentation
User Experience
Security

How do you increase "give a shit" ness?

Introducing Human Driven Development

Human Driven Development acknowledges that software is created by human beings who have emotions and lives outside of the workplace and gives priority to their well-being. Happy developers are productive developers.

Emotions = energy in motion

"The Requirement to FEEL"

Encouraging positive interactions within the team amplifies the overall frequency.

A team with high emotional frequency is more cohesive, motivated, and successful.

Emotional Frequency Chart

	Frequency	Emotional Level	Emotional State	View of Life
Strong Creative Energy	700Hz +	Enlightenment	Inspiration	Is
	600Hz	Peace	Bliss	Perfect
	540Hz	Joy	Serenity	Complete
Negativity dissolves	500Hz	Love	Reverence	Benign
	400Hz	Reason	Understanding	Meaningful
	350Hz	Acceptance	Forgiveness	Harmonious
	310Hz	Willingness	Optimism	Hopeful
	250Hz	Neutrality	Trust	Satisfactory
Energy expands	200Hz	Courage	Affirmation	Feasible
Energy contracts	175Hz	Pride	Scorn	Demanding
	150Hz	Anger	Hate	Antagonistic
	125Hz	Desire	Craving	Disappointing
	100Hz	Fear	Anxiety	Frightening
	75Hz	Grief	Regret	Tragic
Weak Destructive Energy	50Hz	Apathy	Despair	Hopeless
	30Hz	Guilt	Blame	Evil
	20Hz	Shame	Humiliation	Miserable

When your mind is clouded by negative emotions, those frequencies alter what your mind is seeing, which results in you forgetting your wisdom.

Applying HDD in the workplace

"If you are not in the state of either acceptance, enjoyment, or enthusiasm, look closely and you will find that you are creating suffering for yourself and others."

— Eckhart Tolle

Understand what motivates each team member

Example motivation sources: fear, financial gain, camaraderie, challenges

Junior engineers often motivated by fear of making mistakes, fear of speaking up

Fear of deadlines and writing "bad" code can impact motivation

High-vibration motivations (joy, love, passion) lead to better results

Importance of getting to know teammates to boost motivation to support each other

Align work with career goals for better motivation

[promotion] driven development

when you write code to increase your visibility with management, let things break so that you can be the hero that fixes them and generally work on things that lead to a corner office asap.

[fear] driven development

when you attempt to forestall mistakes during coding by moving at a snail's pace, producing, double-checking, and re-triple-checking lots of paper artifacts, before producing any actual working software. In essence, being so afraid of making mistakes that you end up not producing anything at all.

[hate] driven development

when you come to hate working on something so much that it inspires a surge of productivity that leads to completion. Most projects that involve this methodology include a procrastination phase.
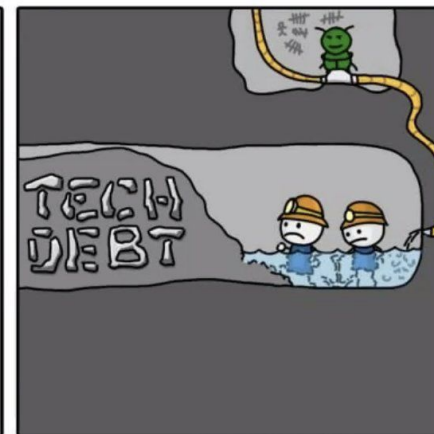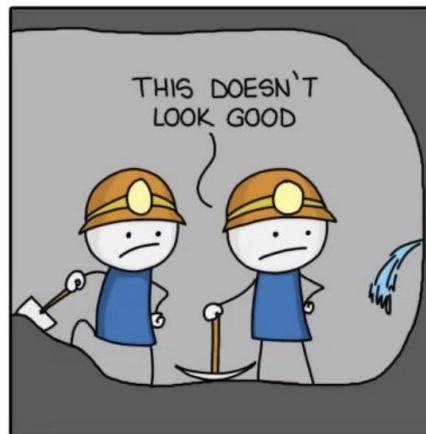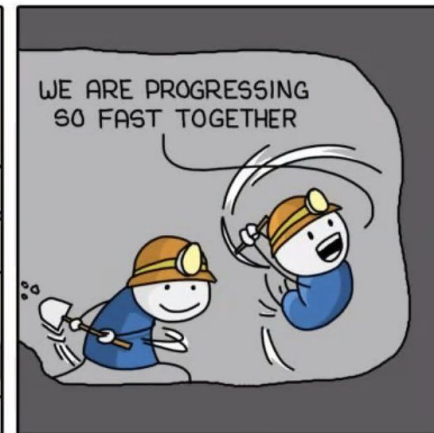
https://devdriven.by/

Pick your battles

Learn to live with tech debt

Balance process with autonomy

Encourage action and decision making

Emphasize pair programming over code reviews.

Real-time feedback and knowledge sharing

Nit-picky code review comments can delay development.

Engineers can feel discouraged by excessive code quality comments.

Pair programming helps engineers learn new things and get real-time feedback.

Senior engineers can quickly identify gaps and improvements.

Junior engineers offer fresh perspectives and innovative insights.

Pairing on work builds camaraderie and rapport, promotes communication, community, and support, and facilitates personal discussions and mental wellness.

Is there a reason for this function to take setter callback arguments rather than just return the billing cycle and member quantity values?

Joey Pedicini

no reason, I'm just burned out and didn't want to change it. I'll change it though

Meetings to foster connection and community

Informal stand-ups over status updates, amplify silent voices in retrospectives

Fri Nov 11 2022

Nov 11, 2022

standups are super uncomfortable because I have no inspiration for what I'm doing and I'm just going through the motions.

So it feels like a lie everytime I talk. I'm not ok but I need the money so I pretend I'm ok.

Thank you!