LEARNING BOOK

Radical Therapy

FOR SOFTWARE DEVELOPMENT TEAMS

Table of Contents

I. Introduction

A. Explanation of Radical Therapy Dev philosophy and how it helps optimize software development teams for better performance and overall happiness
B. Examples of common software development pain points and how Radical Therapy Dev can address them
C. Brief overview of the book's contents and structure

II. Team Dynamics

A. Understanding the importance of team dynamics for the success of software development projects
B. Common challenges faced by software development teams when it comes to team dynamics, including communication gaps and miscommunication, lack of trust, and motivation issues
C. How the Radical Therapy Dev philosophy can help address these challenges and foster a positive team culture
1. Strategies for fostering a growth mindset in team members, including encouraging continuous learning and providing opportunities for career advancement
2. Providing training and education options, such as katas, workshops, and conferences
3. Encouraging individual learning through mentorship and pairing programs
4. Developing a junior and intern program led by mid level and senior developers to provide career advancement opportunities and to help junior team members learn from experienced colleagues.
D. Case studies demonstrating the benefits of fostering a growth mindset in team members
E. Best practices for maintaining a positive team culture and encouraging team members to embrace a growth mindset.

III. Remote Teams

A. Hiring Globally and Thinking Locally
B. Remove the Barrier for Remote Work
C. Tools for Implementing and Policies for sharing

IV. Cloud Native Software Development

A. Definition of cloud-native software development and its benefits for modern software development
B. Strategies for implementing cloud-native software development practices in your team's workflow

C. The role of cloud-native software development in the larger Radical Therapy Dev philosophy

1. How Buildly can be used as a tool to facilitate cloud-native software development
2. Automating infrastructure management with Buildly tools
3. Managing deployments and releases with Buildly Insights
4. Streamlining collaboration and issue tracking with GitHub and Trello integrations

D. Case studies demonstrating how Buildly Insights can be used to manage releases in a cloud-native development environment

E. Step-by-step guide for implementing cloud-native software development practices in your team's workflow, including the use of Buildly Insights.

V. Not so Agile

A. The problem with “agile” and why it isn’t

B. The good things we can learn from it

VI. Radical Process

A. Overview of the Radical Process and its role in the Radical Therapy Dev philosophy

B. Explanation of daily transparency check-ins and how they can benefit software development teams

1. Strategies for implementing the Radical Process in your team's workflow
2. Incorporating daily transparency check-ins into your daily stand-ups
3. Using remote stand-ups to encourage transparency and accountability
4. Creating repeatable processes and flexible frameworks that can be easily adapted to new projects and changing circumstances.

C. Case studies demonstrating the benefits of implementing the Radical Process

D. Best practices for maintaining transparency and accountability in software development teams.

E. Community supported standards for multiple project types

F. The ability to customize a method to fit your needs and share it with others

G. Free and open resources for you to learn and adapt to your needs

VII. Infinite Possibilities. What types of projects can use these methods?

A. Software development practices in the real world

1. Deadline driven projects with tight inflexible timelines
2. Long term maintenance projects with no end in sight.
3. Small to large budget projects with huge project teams or just one person

B. Open Source software or hardware projects with huge distributed project teams

C. Proprietary software projects with small local teams

VIII. How is it different from Agile?

A. Free and open so no expensive experts or books to pay for
B. Adaptable to any project with free examples to build on
C. Use almost any project management tool or build your own open source tool from an open and flexible framework
D. Use Scientific Method rather then Requirements gathering
1. Gather a list of up-front assumptions and list of users to Form a Hypotheses
2. Create simple tests to validate Hypotheses
3. Document results of Hypotheses and make appropriate changes
4. Move on to next stage and continue to test original Hypotheses

IX. Terminology

A. Satellite is an adjustable milestone or marker that communicates to the other satellite and the rest of the project
B. An Orbit is a group of tasks that need to be completed before you reach a satellite.
C. A course correction is one or more changes in priorities or new tasks during an orbit or iteration of a project that can cause satellites to push

out further or come in closer

D. Lifeline is a priority task that other tasks depend on before they can be completed
E. A station is the final satellite or deadline for a project or set of orbits.
F. The event log is a change in course that needs to be documented to show why the change happened and to create an aggregate view of

the decisions or changes

G. Do I have to use this silly space metaphor?
1. Short answer no
2. It can be really helpful, at least the first time, to get people used to the flexible aspect of the methodology

X. Example of a Large distributed Open Source Team

A. Positive Feedback and Achievements
B. Understanding the importance of positive feedback and achievements in motivating software development teams
C. Common mistakes made when providing feedback and how to avoid them
1. Strategies for providing positive feedback and recognition to team members
2. Encouraging team members to recognize each other's achievements
3. Creating a reward system that recognizes both individual and team accomplishments
4. Using data to track team performance and provide feedback

D. Case studies demonstrating the benefits of positive feedback and recognition

E. Best practices for providing effective feedback and creating a culture of recognition in software development teams.’

XI.Conclusions

Chapter 1

Introduction to Radical

The Radical Therapy For Developers Philosophy and How It Can Help

Introduction: Unveiling the Power of Radical Transparency

In the heart of the software industry, a tale of two approaches unfolds – one shrouded in secrecy, the other thriving on openness. The first scenario paints a picture of a proprietary system where decisions were made behind closed doors. A select few dictated the direction, withholding insights from the very users, developers, and managers who would be impacted. This confined strategy seemed enticing in theory, promising control and exclusivity. Yet, as reality set in, the pitfalls emerged. Delays occurred, errors crept in unnoticed, and the end product bore little resemblance to what was needed.

This story of a closed environment resonates with experiences witnessed across industries. From the catastrophic failure of the Theranos scandal to the fiasco of Volkswagen's emissions scandal, hidden intentions often lead to disaster. A study by Transparency International showed that companies with low transparency scores were more prone to unethical behavior and financial misconduct. In software development, this lack of transparency can result in software that fails to meet user needs, leading to dissatisfaction and lost resources.

Contrast this with the rise of open source projects that embody the spirit of collaboration and transparency. The Linux kernel, for instance, is an exemplar of a collaborative approach, fostering innovation through open dialogue and shared decision-making. Studies have shown that open source software exhibits fewer defects and higher quality due to the collective effort of developers worldwide.

What if we could take these lessons and apply them universally in software development? What if we shifted from a secretive proprietary model to one where radical transparency was the cornerstone? Embracing open source principles, this book is your guide to unmasking the potential of radical transparency in software development. By integrating feedback loops, open communication, and inclusive decision-making, we can create software that is not just functional, but adaptive and transformative.

Throughout these pages, we'll explore real-world examples from successful open source projects. We'll examine how companies that practice transparency, have cultivated positive cultures that breed trust and innovation. We'll dissect the shortcomings of proprietary approaches and reveal the practical steps to embrace openness and collaboration. This book is an invitation to rewrite the narrative of software development, product management and organization hierarchies and process in general – one where every stakeholder's voice is heard, decisions are made collectively, and innovation is unbounded by secrecy. Let's embark on this journey to harness the power of Radical Transparency, transforming how we build software and shaping the future of our industry and many others.

Hello and welcome! I am your radical therapist, here to help you bring some much-needed sunshine into the often-cloudy world of software development and cloud native applications. The ideas in here are not just for software but that’s the “use case” we will apply. Any team or set of teams that work from a business side and technical side and have to interact can benefit from this book and the learning within. What I really want you to get out of this though is an idea of how to keep you and the rest of your team, whatever role you play in it, motivated, excited and productive.

Now, I know what you're thinking: "Therapy? For software developers? Isn't that a bit... out there?" But hear me out. In today's fast-paced and constantly-changing world of technology, it's more important than ever to take care of your mental health and well-being. And that's where radical therapy and more importantly radical transparency comes in. By fostering a culture of openness, honesty, and positivity in your team, you can create a work environment where everyone feels empowered to do their best work and grow both personally and professionally. And the best part? You don't have to sacrifice creativity or innovation to do it. In fact most teams see a dramatic increase in productivity as well as increase in the longevity of their teams through the retaining of talent.

In this book, we'll explore the latest research and techniques from the world of modern psychology and combine that with adaptive and iterative processes and tools to help you build a team that's both happy and productive. We'll talk about setting and meeting expectations, fostering a growth mindset, and creating a self-managed development process that allows for flexibility and creativity while still maintaining structure. So buckle up, my dear friends, and get ready for a radical journey to software development nirvana.

Some Tools and an Example
Embracing Radical Transparency in an Example Process

Discover how to integrate Radical Transparency through an illustrative process influenced by successful open-source projects. Dive into a methodology that nurtures open communication, stresses the significance of daily updates, prioritizes thorough documentation and robust testing, encourages early user engagement, fosters transparent code discussions and reviews, aligns with organizational goals, and gains inspiration from real-world instances.

As we start to introduce the concepts and ideas around Radical Transparency, we will use the examples of software development a lot, and specifically we will highlight the successes this process has brought to open source projects, as well as the failure of projects that weren’t transparent enough.

To start, let's go over some needed culture and process tools with cited examples of their successes.

Cultivating Open Communication

Implementing Radical Transparency starts with nurturing an environment of candid communication. Business and technical teams engage in regular virtual stand-up meetings using tools like Slack, making sure that every team member, irrespective of their geographical location, feels equally valued. These stand-ups transcend mere status updates and create an atmosphere where remote and local teams are seamlessly integrated, fostering a unified sense of purpose. The primary goal is to reduce redundant tasks, remove blocks from completing important tasks and to ensure everyone upstream and downstream is aware, but not interfering with the work being done. They should also be done quickly, and facilitate additional discussion where needed.

In the radical transparency version you should focus on what’s next and blocks… not what you did yesterday. This isn’t a competition to see who completed what the fastest, it’s about being on the same page and helping out where we can.

Benefits of Regular Stand-Up Meetings:
- Cross-functional understanding and alignment.
- Early identification and resolution of roadblocks.
- Fostering collaboration and idea exchange.
- Bolstering the sense of a unified team.
- Mitigating potential conflicts through open dialogue.

A study published in the "Journal of Applied Psychology" ([Source: doi.org/10.1037/apl0000220]) highlights that regular stand-up meetings enhance team coordination and mitigate potential conflicts. By providing a platform for open communication, these meetings create an environment where team members feel valued and motivated to contribute effectively.

Daily Summaries and Inclusive Contributions

Central to Radical Transparency is the practice of submitting daily summaries and code contributions. At the close of each workday, team members compile concise summaries of their accomplishments, struggles, and code contributions. This ensures ongoing alignment, prevents bottlenecks, and promotes collaboration.

This is different then the stand up meeting, we focus on what we did and where that work can be found. It's about accountability, removing blocks and celebrating achievements. Kudos.. Or congratulating your team members is especially valued here.

Benefits of Daily Summaries:
- Maintaining an updated overview of ongoing work.
- Addressing potential issues promptly.
- Enhancing accountability and ownership.
- Promoting a shared sense of responsibility.
- Providing a foundation for comprehensive team communication.

A case study conducted by Atlassian ([Source: atlassian.com/blog/agile/daily-standups]) emphasizes the importance of daily summaries. Teams that regularly share their achievements and challenges experience improved collaboration, enhanced productivity, and a heightened sense of mutual support.

Early User Engagement and Feedback Loops

Embrace early user engagement to garner valuable insights that influence the development process. Incorporate the voice of the end user at the outset, using prototypes, mockups, and early-stage versions. This iterative approach helps in delivering user-centric solutions while keeping the team and users on the same page. You need to collect a critical mass of feedback and don't be reactionary. Because one user hates a workflow does not mean they all do, it just means that that user is not afraid to speak up. You should investigate for sure, get additional feedback, then react.

Benefits of Early User Engagement:

Identifying potential design flaws early.
Validating assumptions and minimizing risks.
Enhancing user satisfaction and loyalty.
Establishing clear requirements from the start.
Promoting a collaborative development process.

Transparent Code Discussions and Reviews

Leverage transparent code discussions and reviews to drive quality and accountability. Engage in open conversations around code decisions, providing a platform for diverse opinions. Embrace a culture where every team member's viewpoint is valued and contributes to better code quality.

Code reviews should be seen as part of the workload of every team member who is capable, and should be counted in their proproductivity. A positive comment or suggestion to change an approach or help find a solution can help the team and product save days or even weeks, so encourage and track how much your team is helping each other out.

Benefits of Code Discussions and Reviews:

Enhancing code quality through collective insights.
Reducing the likelihood of overlooked issues.
Fostering a culture of knowledge sharing.
Increasing team morale and satisfaction.
Creating opportunities for skill development and learning.

Alignment with Organizational Goals

Seamlessly align your project's goals with your organization's overarching mission. Incorporate a transparent framework that clearly links the project's objectives with the strategic vision of the company. This approach ensures that every effort made contributes to the organization's long-term success.

Start with a mission and vision statement for your organization if you don’t already have one, and restate the portions of it that match with product mission and vision. If you have Objective and Key Results, or Key Performance Indicators in your organization align those as well with high level features or epics so you can show real measurable results for your organization.

Benefits of Alignment with Organizational Goals:
- Focused efforts that support the company's vision.
- Enhanced decision-making based on strategic alignment.
- Clear understanding of project priorities and outcomes.
- Improved resource allocation and prioritization.
- Engaged teams driven by a shared sense of purpose.

A report by the "Harvard Business Review" emphasized that aligning individual projects with organizational goals ensures efficient resource allocation and strengthens the company's overall strategic direction. Organizations that establish this transparent connection experience improved collaboration and a greater sense of purpose among teams.

Inspiring Transparency from Top to Bottom

Cultivate a culture of transparency that permeates every layer of the organization. Encourage leaders to embody the values of open communication, setting an example for all team members. Establish feedback loops that encourage bottom-up contributions and ensure that all voices are heard and valued.

Create demo days for your project and encourage the rest of your organization to view and comment. Ask them to do the same for their business process or tool and share feedback, ask questions and be a part of the conversation without being disruptive. Create repository of knowledge around each ongoing project or project and allow others access to review and comment.

Benefits of Top-to-Bottom Transparency:
- Increased trust and credibility within the organization.
- Open channels for constructive feedback.
- Fostering innovation through diverse perspectives.
- Boosted employee engagement and satisfaction.
- Alignment of leadership actions with organizational values.

Research conducted by "Gallup" highlighted that organizations with transparent leadership inspire higher levels of engagement and satisfaction among employees. When leaders exemplify openness, employees feel more connected and are motivated to contribute their best work.

A Unified Vision of Success

Forge a cohesive vision that bridges individual efforts with organizational success. Employ alignment tools that visually display how project milestones align with overarching goals. These tools provide a clear pathway for team members to understand how their work directly contributes to the organization's achievements.

Benefits of a Unified Vision:
- Clarified objectives and contributions.
- Enhanced motivation through purposeful work.
- Reduced misalignment and redundancy.
- Increased accountability for project outcomes.
- Stronger commitment to the organization's mission.

In an article published in "Forbes," it was highlighted that organizations using alignment tools to communicate their vision foster a sense of purpose and dedication among employees. This practice enhances collaboration and drives teams to achieve collective success.

The Power of Radical Transparency

Embracing Radical Transparency empowers teams to collaborate openly, enabling swift adaptation, innovation, and alignment with organizational objectives. The approach demonstrated through the open-source model provides insights into building a collaborative, high-performing team that values transparency, inclusivity, and shared success. By cultivating an environment where communication flows freely, where user feedback influences development, and where individual contributions map to broader goals, organizations can transform their projects and cultures. A reporting structure that spans from individual code contributions to company-wide fiscal goals ensures that the benefits of Radical Transparency remain steadfast and continue to drive success.

Therapist 1: "Why does software development have to be so complicated? Isn't writing code, just like writing a joke? Set up, rubber chicken down your pants, then punchline?"
Therapist 2: "Oh, you clearly haven't spent enough time working with software development teams or on your jokes. There are so many moving parts, so many people involved, and everyone has their own unique personality and work style. It's not just about writing code; it's about managing a complex ecosystem of people and processes."
Therapist 1: "I get that, but isn't it just a matter of telling people what to do and then complaining about how lazy everyone is who doesn't get their work done?"
Therapist 2: "Absolutely not! That's a surefire way to demotivate and disengage your team. Effective team dynamics are all about understanding the unique strengths and weaknesses of each team member, creating a supportive and collaborative environment, and providing opportunities for growth and development."
Therapist 1: "But what about negative reinforcement? Surely that's an effective way to get people to do what you want. It works on my cat… No, not in the gift box, in the litter box, Mr. Whiskers!"
Therapist 2: "No, no, no. Negative reinforcement might work in the short term, but it ultimately creates a toxic work environment and drives away top talent. Positive reinforcement, on the other hand, fosters a culture of collaboration and motivation, where team members feel valued and invested in the success of the project."
Therapist 1: "Alright, I see your point. But how do we actually create a positive team dynamic in a software development team?"
Therapist 2: "Great question! It all starts with understanding the unique personalities and work styles of each team member, and creating a supportive and collaborative environment that plays

to their strengths. We can also implement regular check-ins and feedback sessions to ensure that everyone feels heard and valued. And of course, positive reinforcement and recognition for a job well done goes a long way in keeping the team motivated and engaged."

Therapist 1: "I see, I see. So it's not just about writing code, it's about creating a positive team environment that fosters collaboration and success. I guess it's not so simple after all."

Therapist 2: "No, it's definitely not simple. But with the right mindset and approach, we can create a software development dream team that produces amazing results and enjoys doing it."