BUSHIDO DEVOPS

HONOR AND DISCIPLINE MEETS SOFTWARE DISTRIBUTION

WILLIAM KIRK

First Edition

INTRODUCTION

In the fast-paced world of software development and distribution, it's important to use methods that stress both honesty and speed. The Bushido Code's ideas about working together, always getting better, and automating tasks are very similar to DevOps. Both stress discipline, respect, and always trying to be the best.

BUSHIDO TAUGHT the samurai to be very proud of their skills and tasks; they saw their work as an extension of their honor. This same feeling of pride can be seen in the way companies like Toyota treat every step of the process with care and accuracy.

THIS REVERENCE for artistry means that both software development and operations are treated with the same respect in DevOps. Each and every process, release, and line of code must be done perfectly. This dedication to

quality is more than just following technical rules; it's also about respecting the craft of making software.

ACCORDING TO THE BUSHIDO CODE, respect and honor are useful rules that people should follow in their daily lives. In the context of DevOps, this means creating a space where team members respect each other's jobs, talk freely, and work together well. It also means keeping promises about quality, safety, and customer happiness.

BY INCORPORATING Bushido values into DevOps work, an atmosphere of honesty and excellence is created. This way of doing things encourages processes, technologies, and the growth of both people and teams to keep getting better. It makes people feel responsible and accountable, which makes sure that everyone involved in the software lifecycle is dedicated to the greatest standards.

DEVOPS

DevOps is a set of practices, tools, and cultural philosophies that aim to integrate and automate the processes between software development and IT operations. The primary goals are to shorten the development lifecycle, deliver high-quality software continuously, and improve collaboration between teams. Here are key aspects of DevOps:

1. **Collaboration and Communication**: DevOps emphasizes improved collaboration and communication between development (Dev) and operations (Ops) teams. This helps in breaking down silos and fostering a culture of shared responsibility.

2. **Continuous Integration and Continuous Deployment (CI/CD)**: CI/CD practices involve frequently integrating code changes into a shared repository and automating the deployment process. This ensures that software can be released quickly and reliably.

. . .

3. **Automation**: Automating repetitive tasks, such as testing, deployment, and monitoring, reduces the risk of human error and increases efficiency. Automation tools are a core component of DevOps.

4. **Infrastructure as Code (IaC)**: Managing and provisioning computing infrastructure through machine-readable definition files rather than physical hardware configuration or interactive configuration tools. This practice allows for versioning and automation of infrastructure setups.

5. **Monitoring and Logging**: Continuous monitoring and logging of applications and infrastructure help in identifying issues before they affect users. This allows for proactive problem-solving and maintaining system health.

6. **Agility and Continuous Improvement**: DevOps promotes an agile approach to software development and operations, encouraging continuous improvement through feedback loops and iterative processes.

7. **Security (DevSecOps)**: Integrating security practices into the DevOps process ensures that security is considered at every stage of the software development lifecycle, rather than as an afterthought.

. . .

BY ADOPTING DEVOPS, organizations aim to increase the speed and quality of their software delivery while maintaining operational stability and security.

THE BUSHIDO CODE

The Bushido Code, also known as The Way of the Warrior, is a traditional code of conduct and a moral framework that guided the samurai class in feudal Japan. Bushido emphasizes virtues such as loyalty, honor, and discipline. The code has been influential in shaping the ethical and moral values of Japanese culture and continues to be respected and revered. Here are the key principles of the Bushido Code:

1. **Rectitude (義, Gi)**: Moral integrity and righteousness. A samurai must make decisions based on honesty and justice.

2. **Courage (勇, Yū)**: Bravery in the face of danger and adversity. A samurai must show courage in all aspects of life, both physical and moral.

. . .

3. **Benevolence (仁, Jin)**: Kindness and compassion. A samurai should show compassion towards others, especially those who are weaker.

4. **Respect (礼, Rei)**: Politeness and respect towards others. A samurai must treat others with dignity and respect.

5. **Honesty (誠, Makoto)**: Truthfulness and sincerity. A samurai must be honest in words and actions.

6. **Honor (名誉, Meiyo)**: Upholding one's reputation and moral integrity. A samurai must live and die with honor, always striving to maintain a good name.

7. **Loyalty (忠義, Chūgi)**: Devotion and loyalty to one's lord, family, and country. A samurai must be loyal and faithful to those they serve.

8. **Self-Control (自制, Jisei)**: Mastery over one's emotions and desires. A samurai must practice self-discipline and control over their behavior and actions.

THE BUSHIDO CODE was not written as a single document but rather as a collection of values and principles that were passed down through generations of samurai. It

reflects a blend of Confucian, Shinto, and Zen Buddhist influences, shaping the ethical and philosophical outlook of the samurai warrior class.

COMPLEMENTING DEVOPS WITH THE BUSHIDO CODE

Matching the tenets of the Bushido Code with aspects of DevOps can be an insightful exercise, highlighting the ethical and philosophical parallels between the two. Here's how the Bushido Code principles can align with DevOps practices:

1. **Rectitude (義, Gi)** - Moral Integrity and Righteousness - Quality Assurance and Code Integrity: Ensuring that all code meets high standards of quality and is free from defects aligns with the principle of rectitude. This involves rigorous testing and adherence to best practices.

2. **Courage (勇, Yū)** - Bravery - Embracing Change and Innovation: The courage to adopt new technologies, methodologies, and continuous improvement aligns with DevOps. Taking risks to improve processes and deliver value faster embodies this principle.

. . .

3. **Benevolence (仁, Jin)** - Kindness and Compassion - Collaboration and Teamwork: A focus on helping team members, fostering a supportive work environment, and prioritizing the user experience reflects benevolence. DevOps emphasizes empathy towards colleagues and end-users.

4. **Respect (礼, Rei)** - Politeness and Respect - Professionalism and Communication: Treating all team members with respect and maintaining open, respectful communication channels is crucial in DevOps. Respect for different roles and perspectives is essential for collaboration.

5. **Honesty (誠, Makoto)** - Truthfulness and Sincerity - Transparency and Feedback: Being transparent about progress, issues, and mistakes, and providing honest feedback helps build trust within teams. This principle is key to maintaining integrity in DevOps practices.

6. **Honor (名誉, Meiyo)** - Upholding Reputation and Moral Integrity - Commitment to Excellence: Delivering high-quality software and maintaining a good professional reputation aligns with the principle of honor. Consistently meeting or exceeding expectations reflects a commitment to excellence.

7. **Loyalty (忠義, Chūgi)** - Devotion and Loyalty - Commitment to Team and Organization: Demonstrating loyalty to one's team and organization by being dependable

and consistently delivering on commitments is crucial in DevOps. Loyalty fosters trust and long-term success.

8. **Self-Control (自制, Jisei)** - Mastery Over Emotions and Desires - Discipline in Processes and Automation: Practicing self-control in following processes, adhering to standards, and using automation tools to maintain consistency and reliability aligns with this principle. Discipline ensures stability and efficiency in DevOps.

BY ALIGNING THESE PRINCIPLES, both Bushido and DevOps emphasize the importance of ethics, continuous improvement, and a collaborative, respectful approach to achieving excellence.

ONE

CHAPTER 1: RECTITUDE (義, GI) - MORAL INTEGRITY AND RIGHTEOUSNESS

Code pure and steadfast,
Integrity in each line,
Righteous path we walk.

Rectitude, or moral integrity, is a foundational principle of the Bushido Code. It represents the unwavering commitment to righteousness and justice in all actions. For the samurai, rectitude was not merely a guideline but a way of life. It meant making decisions based on what is morally right and standing firm in those decisions, regardless of the consequences. Historical accounts of samurai often highlight their dedication to rectitude, showcasing their willingness to sacrifice personal gain for the greater good.

RECTITUDE IS about living an upright and honorable life. It involves:

- Honesty: Being truthful in all dealings.

- Justice: Ensuring fairness and equity.

- Courage: Standing up for what is right, even in the face of adversity.

- Consistency: Adhering to ethical principles at all times.

ONE NOTABLE EXAMPLE is the story of the Forty-Seven Ronin, a group of samurai who exemplified rectitude through their unwavering loyalty and sense of justice. When their lord was unjustly forced to commit seppuku (ritual suicide), the ronin avenged his death, knowing they would face severe repercussions. Their actions were driven by a deep sense of rectitude, prioritizing their moral duty over their lives.

IN THE CONTEXT OF DEVOPS, rectitude translates to maintaining the highest standards of code quality and ethical conduct. It involves:

- Ensuring Code Quality and Integrity: Writing clean, efficient, and maintainable code. This requires adherence to coding standards and best practices, rigorous code reviews, and continuous testing.

- Adherence to Best Practices and Ethical Standards: Following industry standards and guidelines, ensuring compliance with legal and ethical requirements. This includes respecting user privacy, securing sensitive data, and maintaining transparency.

- Making Ethical Decisions: Prioritizing the long-term health of the codebase and the organization over short-term gains. This might involve refactoring legacy code, resisting the temptation to cut corners, and advo-

cating for necessary changes even when they are unpopular.

IMPLEMENTATIONS:

Google's Code of Conduct: Google's emphasis on the principle Don't be evil reflects a commitment to rectitude. This principle guides employees to act with integrity and prioritize user trust and security in their work.

ETSY'S DEPLOYMENT PIPELINE: Etsy's focus on continuous deployment and rigorous automated testing ensures that every code change meets high standards of quality and integrity before it reaches production. This practice reflects a deep commitment to rectitude, ensuring that users receive a reliable and consistent experience.

RECTITUDE, as a principle of the Bushido Code, offers valuable guidance for modern DevOps practices. By prioritizing moral integrity and righteousness, teams can ensure high-quality code, ethical behavior, and long-term success. As we move forward, integrating rectitude into our daily work can help us build a more trustworthy and resilient software development ecosystem.

CHAPTER 1: EXERCISES

Exercise 1: Code Quality Audit

Objective: Improve code integrity by identifying and addressing areas of improvement.

INSTRUCTIONS:

1. Select a recent codebase or project.
2. Conduct a thorough code review, focusing on quality and adherence to best practices.
3. Identify areas where the code deviates from established standards or contains potential issues.
4. Document your findings and provide constructive feedback.
5. Implement necessary changes to improve code quality.

OUTCOME: Enhanced understanding of code quality standards and improved code integrity.

Exercise 2: Ethical Dilemma Discussion

Objective: Foster ethical decision-making in development processes.

INSTRUCTIONS:

1. Form small groups of 3-5 team members.
2. Present an ethical dilemma related to software development (e.g., handling user data, shortcuts in security protocols).
3. Discuss the dilemma and explore potential solutions, considering the principles of rectitude.
4. Each group presents their proposed solution and rationale.
5. Reflect on how ethical principles guide decision-making in real-world scenarios.

OUTCOME: Enhanced ability to navigate ethical challenges and make decisions based on integrity.

Exercise 3: Continuous Integration Implementation

Objective: Ensure code integrity through continuous integration (CI) practices.

INSTRUCTIONS:

1. Set up a CI pipeline for a project using tools like Jenkins, Travis CI, or GitHub Actions.
2. Configure the pipeline to run automated tests and code quality checks on each commit.
3. Ensure the pipeline includes steps for linting, unit tests, and integration tests.
4. Monitor the pipeline and address any issues that arise.
5. Reflect on how CI practices contribute to maintaining code integrity.

OUTCOME: Practical experience with CI tools and an understanding of how they support code quality and integrity.

Exercise 4: Creating a Code of Conduct

Objective: Establish guidelines for maintaining integrity and ethical behavior within the team.

INSTRUCTIONS:

1. Collaborate with your team to draft a code of conduct.
2. Define key principles and behaviors that align with rectitude and ethical standards.
3. Include guidelines for code quality, collaboration, communication, and decision-making.
4. Review and finalize the code of conduct with input from all team members.

5. Implement the code of conduct and regularly review its effectiveness.

OUTCOME: A shared understanding of ethical standards and a formalized code of conduct to guide team behavior.

Exercise 5: Retrospective on Past Projects

Objective: Reflect on past projects to identify areas where rectitude was upheld or could be improved.

INSTRUCTIONS:

1. Choose a completed project for review.
2. Conduct a retrospective meeting with the team.
3. Discuss instances where decisions and actions aligned with the principles of rectitude.
4. Identify situations where integrity may have been compromised and explore ways to improve.
5. Document lessons learned and create an action plan for future projects.

OUTCOME: Increased awareness of how rectitude influences project outcomes and actionable insights for maintaining integrity in future work.

TWO

CHAPTER 2: COURAGE (勇, YŪ) - BRAVERY

Change embraced with strength,
Innovation lights the way,
Fearless hearts prevail.

Courage, or Yū, is another core principle of the Bushido Code. It represents bravery, not just in the face of physical danger but also in moral and ethical challenges. For the samurai, courage was about facing fear and adversity with strength and determination. This principle is reflected in many historical accounts where samurai displayed remarkable bravery in battle and in making difficult decisions that required moral fortitude.

THE ESSENCE of Courage in Bushido

Courage involves:

- Facing Fear: Confronting fear head-on rather than avoiding it.

- Taking Risks: Willingness to take calculated risks for the sake of duty or a higher cause.

- Moral Bravery: Standing up for what is right, even when it is unpopular or difficult.

- Resilience: Persisting in the face of setbacks and challenges.

STORIES AND EXAMPLES of Courage in Samurai History

One of the most famous examples of samurai courage is the story of Minamoto no Yoshitsune. Despite being vastly outnumbered, Yoshitsune led his troops to victory in several critical battles during the Genpei War. His strategic brilliance and unwavering courage in the face of overwhelming odds made him a legendary figure in Japanese history.

APPLYING Courage in DevOps

In the context of DevOps, courage translates to embracing change, innovation, and continuous improvement. It involves:

- Embracing Change and Innovation: Being open to adopting new technologies and methodologies. This can mean transitioning from traditional practices to agile and DevOps practices, which require a shift in mindset and processes.

- Overcoming Fear of Failure: In DevOps, failure is seen as an opportunity to learn and improve. Encouraging a culture where team members feel safe to take risks and experiment is essential.

- Taking Calculated Risks: Implementing new tools or

processes involves risks, but with proper planning and testing, these risks can lead to significant improvements in efficiency and quality.

- Resilience and Persistence: Continuously working towards improvement, even when faced with setbacks or challenges. This involves learning from failures and using those lessons to drive future success.

IMPLEMENTATIONS

Netflix's Chaos Engineering: Netflix's approach to resilience, known as Chaos Engineering, involves deliberately introducing failures into their system to test its robustness. This practice requires courage as it involves taking risks to ensure that the system can withstand real-world failures. By doing so, Netflix has built one of the most resilient streaming platforms in the world.

ETSY'S MOVE to Continuous Deployment: Etsy's transition to continuous deployment required significant changes in their development and deployment processes. The team embraced this change courageously, despite the risks involved. The result was a more agile and responsive development cycle, allowing Etsy to deploy code changes quickly and reliably.

COURAGE, as a principle of the Bushido Code, is essential for modern DevOps practices. By embracing change, innovation, and continuous improvement, teams can achieve greater efficiency, resilience, and quality. As we move

forward, integrating courage into our daily work can help us build a more dynamic and successful software development ecosystem.

CHAPTER 2: EXERCISES

Exercise 1: Embrace a New Technology

Objective: Encourage the team to explore and adopt a new technology or tool.

INSTRUCTIONS:

1. Identify a new technology or tool relevant to your projects (e.g., a new programming language, framework, or DevOps tool).
2. Form small groups to research and experiment with the technology.
3. Have each group create a small project or proof of concept using the new technology.
4. Present findings and experiences to the team, highlighting challenges and benefits.
5. Discuss how embracing new technologies requires courage and can lead to innovation.

. . .

OUTCOME: Increased familiarity with new technologies and a culture of experimentation and innovation.

Exercise 2: Failure Stories Sharing Session

Objective: Normalize the concept of failure as a learning opportunity.

INSTRUCTIONS:

1. Schedule a team meeting focused on sharing personal or professional experiences with failure.
2. Encourage team members to share stories about a time they failed and what they learned from it.
3. Discuss how these experiences contributed to personal and professional growth.
4. Reflect on the importance of courage in facing and learning from failures.
5. Document key takeaways and strategies for handling future challenges.

OUTCOME: A supportive environment that values learning from failure and promotes resilience and courage.

Exercise 3: Innovation Challenge

Objective: Foster a culture of innovation by encouraging team members to propose and implement new ideas.

. . .

INSTRUCTIONS:

1. Announce an innovation challenge, inviting team members to propose new ideas or improvements for current processes or products.

2. Form teams to develop and implement the most promising ideas over a set period (e.g., two weeks).

3. Provide resources and support to help teams execute their projects.

4. Host a presentation session where each team showcases their innovation and discusses the outcomes.

5. Celebrate the efforts and courage to innovate, regardless of the results.

OUTCOME: Enhanced innovation and a culture that encourages taking risks and exploring new ideas.

Exercise 4: Incident Response Simulation

Objective: Build confidence and skills in handling unexpected challenges and crises.

INSTRUCTIONS:

1. Plan a simulated incident response exercise (e.g., a system outage or security breach).

2. Assign roles and responsibilities to team members for handling the incident.

3. Run the simulation, guiding the team through identi-

fying, diagnosing, and resolving the issue.

4. Conduct a post-incident review to discuss what went well and areas for improvement.

5. Reflect on the importance of maintaining courage and composure during real incidents.

OUTCOME: Improved incident response skills and confidence in handling crises with bravery and efficiency.

Exercise 5: Personal Development Plan

Objective: Encourage team members to set and pursue challenging personal and professional goals.

INSTRUCTIONS:

1. Have each team member create a personal development plan outlining specific goals they want to achieve.

2. Goals should include both short-term and long-term objectives, such as learning a new skill, obtaining a certification, or contributing to an open-source project.

3. Schedule regular check-ins to discuss progress, challenges, and achievements.

4. Provide support and resources to help team members reach their goals.

5. Celebrate milestones and reflect on how pursuing challenging goals requires and builds courage.

. . .

OUTCOME: Personal and professional growth for team members and a culture of continuous improvement and bravery in pursuing new challenges.

THREE

CHAPTER 3: BENEVOLENCE (仁, JIN) - KINDNESS AND COMPASSION

Kindness in our code,
Teamwork builds a better world,
Compassion in sync.

Benevolence, or Jin, is a central tenet of the Bushido Code, emphasizing kindness and compassion. For the samurai, benevolence was about using their strength and skills to protect and help others, rather than for personal gain. This principle was crucial in maintaining social harmony and ensuring the well-being of those under their care.

THE ESSENCE of Benevolence in Bushido

Benevolence involves:

- Kindness: Showing compassion and understanding towards others.
- Generosity: Willingness to give and support without expecting anything in return.

- Empathy: Understanding and sharing the feelings of others.

- Protection: Using one's abilities to safeguard others and ensure their well-being.

STORIES AND EXAMPLES of Benevolence in Samurai History

A famous example of benevolence is the story of Mori Motonari, a samurai lord known for his compassionate leadership. Motonari prioritized the welfare of his people, implementing policies that improved their lives. His benevolence earned him the loyalty and respect of his subjects, leading to a stable and prosperous domain.

APPLYING Benevolence in DevOps

In the context of DevOps, benevolence translates to fostering collaboration, teamwork, and a user-centric approach. It involves:

- Fostering Collaboration and Teamwork: Encouraging a supportive and cooperative work environment where team members help each other and share knowledge.

- Prioritizing User Experience and Customer Satisfaction: Ensuring that the needs and expectations of users are at the forefront of development and operations. This involves creating user-friendly, reliable, and high-quality software.

- Empathy and Understanding: Developing a deep understanding of the challenges and needs of both team members and users. This can lead to more effective problem-solving and innovation.

- Support and Mentorship: Providing guidance and

support to team members, especially those who are less experienced. This helps in building a stronger and more cohesive team.

IMPLEMENTATIONS

Amazon's Customer Obsession: Amazon's relentless focus on customer satisfaction is a prime example of benevolence in practice. By prioritizing the needs of customers and continuously improving the user experience, Amazon has built a loyal customer base and a successful business model.

SPOTIFY'S SQUAD MODEL: Spotify's organizational structure, known as the squad model, emphasizes teamwork and collaboration. Squads are small, cross-functional teams that work together towards common goals, fostering a supportive environment where team members can thrive.

BENEVOLENCE, as a principle of the Bushido Code, offers valuable guidance for modern DevOps practices. By fostering collaboration, prioritizing user experience, and showing empathy and kindness, teams can create high-quality software and build a positive work environment. As we move forward, integrating benevolence into our daily work can help us build a more compassionate and successful software development ecosystem.

CHAPTER 3: EXERCISES

Exercise 1: Peer Support Initiative

Objective: Foster a culture of mutual support and collaboration within the team.

INSTRUCTIONS:

1. Pair team members into peer support partners.
2. Encourage each pair to schedule regular check-ins (e.g., weekly or bi-weekly) to discuss their work, challenges, and provide support.
3. During check-ins, partners should offer advice, help troubleshoot issues, and provide encouragement.
4. Rotate peer support partners every few months to build relationships across the team.
5. Reflect on how supporting one another fosters a compassionate and collaborative work environment.

. . .

OUTCOME: Enhanced team cohesion, mutual support, and a stronger sense of community.

Exercise 2: Empathy Mapping for Users

Objective: Deepen understanding of user needs and experiences through empathy.

INSTRUCTIONS:

1. Introduce the concept of empathy mapping to the team.
2. Create empathy maps for different user personas, focusing on what users think, feel, say, and do.
3. Identify user pain points, needs, and desires.
4. Discuss how insights from the empathy maps can inform product development and improve user experience.
5. Implement changes based on the empathy maps and monitor their impact.

OUTCOME: Improved user experience and products that better meet user needs through a compassionate approach.

Exercise 3: Random Acts of Kindness

Objective: Encourage acts of kindness within the team to build a positive and supportive environment.

. . .

INSTRUCTIONS:

1. Encourage team members to perform random acts of kindness for their colleagues (e.g., helping with a task, providing a compliment, or sharing a resource).

2. Create a Kindness Board where team members can anonymously post notes about acts of kindness they have received or observed.

3. Recognize and celebrate these acts during team meetings.

4. Reflect on how these acts contribute to a more positive and supportive team culture.

5. Encourage ongoing participation in random acts of kindness.

OUTCOME: A more positive and supportive work environment where kindness and compassion are actively practiced.

Exercise 4: Team Building Through Volunteering

Objective: Strengthen team bonds and foster compassion by participating in volunteer activities.

INSTRUCTIONS:

1. Organize a team volunteer day with a local charity or community organization.

2. Choose activities that allow the team to work together

and contribute to a meaningful cause (e.g., a food bank, animal shelter, or environmental cleanup).

3. Reflect on the experience as a team, discussing how volunteering has impacted their perspective and strengthened their bonds.

4. Encourage ongoing participation in volunteer activities, either individually or as a team.

5. Share the positive impacts of volunteering within the organization to inspire others.

OUTCOME: Stronger team bonds, a sense of fulfillment, and a culture of compassion and social responsibility.

Exercise 5: Feedback with Compassion

Objective: Improve the quality and impact of feedback by incorporating empathy and compassion.

INSTRUCTIONS:

1. Train team members on how to provide constructive feedback using compassionate language.

2. Practice giving and receiving feedback in pairs, focusing on empathy and understanding.

3. Use role-playing scenarios to simulate difficult feedback situations and practice responding with compassion.

4. Implement a feedback system that encourages regular, compassionate feedback exchanges.

5. Reflect on the impact of compassionate feedback on team dynamics and individual growth.

. . .

OUTCOME: Enhanced feedback culture that promotes personal and professional growth through empathy and understanding.

FOUR

CHAPTER 4: RESPECT (礼, REI) - POLITENESS AND RESPECT

Respect guides our hands,
Politeness in every task,
Harmony in work.

Respect, or Rei, is a fundamental principle of the Bushido Code, emphasizing politeness and respect towards others. For the samurai, respect was not just about manners, but about recognizing the inherent dignity and worth of all individuals. This principle was evident in their interactions, both in times of peace and conflict, and was essential for maintaining harmony within society.

THE ESSENCE of Respect in Bushido

Respect involves:

- Politeness: Demonstrating good manners and courteous behavior.

- Recognition of Dignity: Acknowledging the value and worth of every person.

- Humility: Being modest and humble, regardless of one's status or achievements.

- Harmony: Promoting peaceful and harmonious relationships.

STORIES AND EXAMPLES of Respect in Samurai History

A notable example of respect in samurai history is the practice of bowing before and after a duel. Regardless of the outcome, samurai would show respect to their opponent, recognizing their skill and bravery. This practice underscored the importance of maintaining dignity and honor, even in conflict.

APPLYING Respect in DevOps

In the context of DevOps, respect translates to maintaining professionalism, effective communication, and valuing diverse perspectives. It involves:

- Maintaining Professionalism and Effective Communication: Ensuring all interactions are conducted with courtesy and respect. This includes clear and respectful communication, especially during code reviews, meetings, and collaborations.

- Valuing Diverse Roles and Perspectives: Recognizing and appreciating the contributions of all team members, regardless of their role. This fosters a more inclusive and innovative environment.

- Encouraging Humility: Acknowledging that no one person has all the answers and being open to learning from

others. This helps create a culture of continuous improvement.

- Promoting Harmony and Collaboration: Creating a work environment where team members feel safe and respected, leading to better collaboration and productivity.

IMPLEMENTATIONS

Google's Code Review Culture: Google's approach to code reviews emphasizes respect and constructive feedback. Reviewers provide feedback in a respectful manner, focusing on improving the code rather than criticizing the coder. This fosters a culture of learning and mutual respect.

MICROSOFT'S GROWTH MINDSET: Under Satya Nadella's leadership, Microsoft adopted a growth mindset culture that values learning and humility. This shift has promoted a more respectful and collaborative work environment, driving innovation and success.

RESPECT, as a principle of the Bushido Code, is essential for modern DevOps practices. By maintaining professionalism, valuing diverse perspectives, and promoting a harmonious work environment, teams can achieve greater collaboration, innovation, and success. As we move forward, integrating respect into our daily work can help us build a more respectful and effective software development ecosystem.

CHAPTER 4: EXERCISES

Exercise 1: Active Listening Workshop

Objective: Improve communication skills by practicing active listening and demonstrating respect for others' viewpoints.

INSTRUCTIONS:

1. Conduct a workshop on active listening techniques, including maintaining eye contact, not interrupting, and paraphrasing what the speaker has said.
2. Pair team members and have them take turns speaking and listening on various topics.
3. After each session, have the listener summarize what they heard and confirm with the speaker.
4. Reflect on how active listening can enhance respect and understanding within the team.
5. Encourage team members to practice active listening in all meetings and interactions.

. . .

OUTCOME: Improved communication and a deeper sense of respect and understanding among team members.

Exercise 2: Cultural Appreciation Day

Objective: Foster respect and appreciation for cultural diversity within the team.

INSTRUCTIONS:

1. Organize a Cultural Appreciation Day where team members can share aspects of their cultural backgrounds, such as traditions, holidays, food, and customs.
2. Encourage presentations, demonstrations, and discussions to celebrate and learn about each other's cultures.
3. Create a cultural potluck lunch where team members bring a dish representative of their culture.
4. Reflect on the importance of respecting and valuing cultural diversity in the workplace.
5. Discuss ways to incorporate cultural awareness and respect into daily team interactions.

OUTCOME: Increased cultural awareness, respect for diversity, and a stronger sense of community within the team.

Exercise 3: Respectful Communication Guidelines

Objective: Establish and implement guidelines for maintaining respectful communication within the team.

INSTRUCTIONS:

1. Collaborate with the team to create guidelines for respectful communication, including language use, tone, and behavior during meetings and discussions.
2. Discuss the importance of these guidelines and how they contribute to a respectful work environment.
3. Post the guidelines in common areas and include them in onboarding materials for new team members.
4. Regularly review and update the guidelines to ensure they remain relevant and effective.
5. Reflect on how following these guidelines has impacted team interactions and respect.

OUTCOME: Clear and consistent communication standards that promote respect and positive interactions within the team.

Exercise 4: Peer Recognition Program

Objective: Encourage team members to recognize and appreciate each other's contributions and efforts.

INSTRUCTIONS:

1. Implement a peer recognition program where team members can nominate colleagues for acts of respect, kindness, and collaboration.

2. Set up a recognition board or digital platform where nominations and appreciations can be shared.

3. Regularly celebrate recognized team members during meetings or through internal communications.

4. Encourage team members to provide specific examples of how their colleagues demonstrated respect or went above and beyond.

5. Reflect on how the recognition program has fostered a culture of respect and appreciation.

OUTCOME: Increased morale and a culture of mutual respect and recognition within the team.

Exercise 5: Conflict Resolution Training

Objective: Equip team members with skills to manage and resolve conflicts respectfully and effectively.

INSTRUCTIONS:

1. Conduct a training session on conflict resolution techniques, including understanding different conflict styles, active listening, and finding common ground.

2. Role-play common conflict scenarios and practice using the techniques learned.

3. Discuss the importance of addressing conflicts early and respectfully to maintain a positive work environment.

4. Provide resources and support for team members to apply conflict resolution skills in real situations.

5. Reflect on how effectively resolving conflicts can enhance respect and collaboration within the team.

OUTCOME: Improved conflict resolution skills and a more respectful and harmonious work environment.

FIVE

CHAPTER 5: HONESTY (誠, MAKOTO) - TRUTHFULNESS AND SINCERITY

Truth in every word,
Open feedback shapes our path,
Honesty prevails.

Honesty, or Makoto, is a core tenet of the Bushido Code, emphasizing truthfulness and sincerity in all actions and communications. For the samurai, honesty was about living an authentic life, being true to oneself and others. It was crucial for building trust and maintaining one's honor and reputation.

THE ESSENCE of Honesty in Bushido

Honesty involves:

- Truthfulness: Being honest in words and actions.
- Sincerity: Acting with genuine intentions and integrity.

- Transparency: Being open and clear about one's actions and decisions.

- Trustworthiness: Building and maintaining trust through consistent honesty.

STORIES AND EXAMPLES of Honesty in Samurai History

One famous example of samurai honesty is the story of Takeda Shingen, a samurai warlord known for his integrity. When Shingen discovered that one of his men had stolen from a villager, he ordered the man to return the stolen goods and apologize. This act demonstrated Shingen's commitment to honesty and justice, reinforcing the trust and respect of his people.

APPLYING Honesty in DevOps

In the context of DevOps, honesty translates to ensuring transparency, providing open feedback, and maintaining ethical standards. It involves:

- Ensuring Transparency and Open Feedback: Creating a culture where team members feel safe to speak openly about issues and provide constructive feedback. This leads to better decision-making and problem-solving.

- Maintaining Ethical Standards: Adhering to ethical practices in all aspects of work, from coding and testing to deployment and operations. This includes respecting user privacy, securing sensitive data, and avoiding shortcuts that compromise quality.

- Building Trust Through Consistent Honesty: Being reliable and trustworthy in all interactions. This fosters a culture of mutual respect and collaboration.

. . .

IMPLEMENTATIONS

Google's Blameless Postmortems: Google's practice of conducting blameless postmortems after incidents promotes honesty and transparency. Team members can openly discuss what went wrong and how to prevent it in the future without fear of blame or retribution. This approach encourages learning and continuous improvement.

ETSY'S COMMITMENT TO TRANSPARENCY: Etsy is known for its commitment to transparency, both internally and externally. The company shares detailed reports on system outages, including what happened, why it happened, and how they plan to prevent future occurrences. This openness builds trust with users and employees alike.

HONESTY, as a principle of the Bushido Code, is vital for modern DevOps practices. By ensuring transparency, providing open feedback, and maintaining ethical standards, teams can build trust, foster collaboration, and achieve greater success. As we move forward, integrating honesty into our daily work can help us build a more trustworthy and effective software development ecosystem.

CHAPTER 5: EXERCISES

Exercise 1: Transparency in Projects

Objective: Foster a culture of openness and honesty by sharing project details and progress.

INSTRUCTIONS:

1. Schedule regular project update meetings where team members present their progress, challenges, and plans.
2. Encourage open discussions about any obstacles or issues faced during the project.
3. Share project timelines, goals, and key performance indicators (KPIs) with the entire team.
4. Reflect on how transparency helps build trust and ensures everyone is aligned.
5. Document the outcomes of these meetings and track progress against the goals.

. . .

OUTCOME: Improved project transparency and a culture of openness and trust within the team.

Exercise 2: Honest Retrospectives

Objective: Promote a culture of continuous improvement through honest reflection on past projects or sprints.

INSTRUCTIONS:

1. Conduct regular retrospective meetings at the end of each project or sprint.
2. Use a structured format such as Start-Stop-Continue to discuss what went well, what didn't, and what can be improved.
3. Encourage team members to be honest and constructive in their feedback.
4. Create action items based on the discussion and assign responsibility for each.
5. Follow up on action items in subsequent retrospectives to ensure continuous improvement.

OUTCOME: Enhanced ability to learn from past experiences and a culture of honesty and continuous improvement.

Exercise 3: Ethical Decision-Making Scenarios

Objective: Enhance ethical awareness and decision-making skills through practical scenarios.

INSTRUCTIONS:

1. Present the team with various ethical dilemmas related to software development (e.g., handling user data, security vulnerabilities, feature prioritization).
2. Divide the team into small groups and have each group discuss and decide on the best course of action for each scenario.
3. Each group presents their decisions and rationale to the larger team.
4. Discuss the importance of honesty and ethical behavior in each scenario.
5. Reflect on how these exercises can guide ethical decision-making in real-world situations.

OUTCOME: Improved ethical decision-making skills and a deeper understanding of the importance of honesty in the workplace.

Exercise 4: Honest Feedback Sessions

Objective: Create a culture of honesty and growth through regular feedback sessions.

. . .

INSTRUCTIONS:

1. Pair team members and schedule regular one-on-one feedback sessions.

2. Encourage team members to provide honest, constructive feedback focused on specific behaviors and outcomes.

3. Use a feedback framework such as SBI (Situation-Behavior-Impact) to structure feedback.

4. Reflect on the feedback received and identify areas for personal and professional growth.

5. Follow up on feedback in subsequent sessions to track progress and improvements.

OUTCOME: Enhanced communication, personal growth, and a culture of honesty and trust.

Exercise 5: Code Transparency and Documentation

Objective: Ensure code integrity and transparency through comprehensive documentation and sharing.

INSTRUCTIONS:

1. Establish a documentation standard for all code, including comments, readme files, and architectural diagrams.

2. Encourage team members to document their code thoroughly and keep documentation up to date.

3. Conduct regular code review sessions where team members present and explain their code to others.

4. Use these sessions to provide honest feedback and suggestions for improvement.

5. Reflect on how thorough documentation and transparent code practices enhance trust and collaboration.

OUTCOME: Improved code quality, transparency, and a culture of honesty and accountability in coding practices.

SIX

CHAPTER 6: HONOR (名誉, MEIYO) - UPHOLDING REPUTATION AND MORAL INTEGRITY

Honor in our craft,
Excellence we strive to reach,
Reputation earned.

Honor, or Meiyo, is a central principle of the Bushido Code, emphasizing the importance of maintaining one's reputation and moral integrity. For the samurai, honor was a measure of their worth and a reflection of their character. It was essential to live and act in a way that upheld their honor and the honor of their family and lord.

THE ESSENCE of Honor in Bushido

Honor involves:

- Reputation: Building and maintaining a good name through virtuous actions.
- Moral Integrity: Acting in accordance with ethical principles and values.

- Respect for Self and Others: Upholding one's dignity and the dignity of others through honorable behavior.

- Accountability: Taking responsibility for one's actions and their consequences.

STORIES AND EXAMPLES of Honor in Samurai History

A well-known example of honor in samurai history is the tale of Miyamoto Musashi, a legendary swordsman and ronin. Musashi's life was marked by his relentless pursuit of excellence and his adherence to the samurai code of honor. His duels and battles were fought with a deep sense of responsibility and respect for his opponents, solidifying his reputation as an honorable warrior.

APPLYING Honor in DevOps

In the context of DevOps, honor translates to a commitment to excellence, ethical conduct, and accountability. It involves:

- Commitment to Excellence: Striving to deliver high-quality software and uphold the highest standards in all aspects of development and operations.

- Ethical Conduct: Acting with integrity and honesty, and ensuring that decisions and actions align with ethical principles.

- Accountability: Taking responsibility for one's work, including successes and failures, and learning from mistakes to continuously improve.

- Respect for Colleagues and Stakeholders: Treating all team members, users, and stakeholders with respect and dignity.

. . .

IMPLEMENTATIONS

GitHub's Open Source Community: GitHub fosters a culture of honor and respect within its open source community. Contributors are encouraged to adhere to a code of conduct that emphasizes respectful and ethical behavior, ensuring a positive and productive environment for collaboration.

MICROSOFT'S CODE OF CONDUCT: Microsoft's commitment to ethical conduct and accountability is reflected in its comprehensive code of conduct. This code guides employees in upholding the company's values and maintaining its reputation for integrity and excellence.

HONOR, as a principle of the Bushido Code, is essential for modern DevOps practices. By committing to excellence, ethical conduct, and accountability, teams can build trust, foster collaboration, and achieve greater success. As we move forward, integrating honor into our daily work can help us build a more reputable and effective software development ecosystem.

CHAPTER 6: EXERCISES

Exercise 1: Excellence in Code Reviews

Objective: Uphold high standards of code quality and integrity through thorough and respectful code reviews.

INSTRUCTIONS:

1. Pair team members for code reviews and schedule regular review sessions.
2. Establish clear guidelines for providing constructive and respectful feedback.
3. Focus on identifying areas for improvement and upholding coding standards.
4. Encourage reviewers to recognize and praise excellent code.
5. Reflect on how maintaining high standards in code reviews contributes to the team's overall honor and reputation.

. . .

OUTCOME: Improved code quality, a culture of respect and excellence, and enhanced team reputation.

Exercise 2: Honor Code Creation

Objective: Develop a team honor code that outlines the principles and behaviors expected of all team members.

INSTRUCTIONS:

1. Organize a brainstorming session to discuss the values and principles that are important to the team.
2. Collaborate to draft a team honor code that includes commitments to excellence, integrity, accountability, and respect.
3. Ensure the honor code is visible to all team members and included in onboarding materials.
4. Regularly review and update the honor code to ensure it remains relevant.
5. Reflect on how adhering to the honor code enhances the team's reputation and moral integrity.

OUTCOME: A clear set of principles guiding team behavior and a stronger sense of shared values and honor.

Exercise 3: Ethical Case Studies

Objective: Discuss real-world ethical dilemmas to reinforce the importance of maintaining honor and integrity.

INSTRUCTIONS:

1. Select real-world case studies involving ethical dilemmas in the tech industry (e.g., data privacy issues, security breaches, ethical AI use).
2. Divide the team into small groups and have each group analyze a case study.
3. Discuss the ethical challenges presented and potential solutions, emphasizing the importance of honor and integrity.
4. Each group presents their analysis and proposed solutions to the larger team.
5. Reflect on how these discussions can guide ethical decision-making and uphold the team's honor.

OUTCOME: Enhanced ethical awareness and decision-making skills, reinforcing the importance of honor and integrity.

Exercise 4: Personal Accountability Pledge

Objective: Foster a culture of accountability and personal responsibility within the team.

. . .

INSTRUCTIONS:

1. Have each team member write a personal accountability pledge outlining their commitment to honor, integrity, and excellence.

2. Encourage team members to share their pledges with the team and discuss how they plan to uphold these commitments.

3. Create a system for regular check-ins to discuss progress and challenges in adhering to the pledges.

4. Recognize and celebrate team members who consistently demonstrate accountability and honor.

5. Reflect on how personal accountability contributes to the team's overall reputation and success.

OUTCOME: A stronger sense of personal responsibility and a culture of accountability and honor within the team.

Exercise 5: Recognition of Excellence

Objective: Regularly recognize and celebrate instances of excellence and honorable behavior within the team.

INSTRUCTIONS:

1. Establish a recognition program to highlight team members who demonstrate excellence, integrity, and honor in their work.

2. Encourage team members to nominate colleagues who have upheld the team's values and made significant contributions.

3. Celebrate recognized individuals during team meetings or through internal communications.

4. Reflect on the impact of recognizing and celebrating excellence on team morale and culture.

5. Ensure the recognition program is inclusive and consistently applied.

OUTCOME: Increased motivation, morale, and a culture that values and upholds honor and excellence.

SEVEN

CHAPTER 7: LOYALTY (忠義, CHŪGI) - DEVOTION AND LOYALTY

Loyal to the team,
Commitment through thick and thin,
Trust and faith endure.

Loyalty, or Chūgi, is a fundamental principle of the Bushido Code, emphasizing unwavering devotion and loyalty. For the samurai, loyalty was about being steadfastly committed to their lord, family, and comrades. This principle was crucial for maintaining trust and cohesion within the samurai class and their communities.

THE ESSENCE of Loyalty in Bushido

Loyalty involves:

- Devotion: Being dedicated and faithful to one's duties and relationships.
- Trustworthiness: Being reliable and dependable in all situations.

- Commitment: Demonstrating steadfastness and perseverance.

- Sacrifice: Being willing to put the needs of others before oneself.

STORIES AND EXAMPLES of Loyalty in Samurai History

The story of the Forty-Seven Ronin exemplifies the principle of loyalty. After their lord was forced to commit seppuku due to a political dispute, the ronin avenged his death, knowing it would lead to their own demise. Their actions demonstrated unwavering loyalty to their lord and each other, becoming a legendary tale of devotion and honor in Japanese history.

APPLYING Loyalty in DevOps

In the context of DevOps, loyalty translates to commitment to team and organizational goals, fostering trust, and supporting one another. It involves:

- Demonstrating Commitment to Team and Organization: Being dedicated to the success of the team and the organization. This means aligning personal goals with team objectives and consistently contributing to shared goals.

- Fostering Trust and Dependability: Building a culture of reliability where team members can depend on each other. This involves clear communication, meeting commitments, and supporting colleagues.

- Supporting One Another: Providing help and encouragement to team members, especially during challenging times. This creates a supportive and resilient team environment.

- Aligning with Organizational Values and Goals: Ensuring that personal and team efforts are aligned with the broader mission and values of the organization.

IMPLEMENTATIONS

Netflix's Culture of Freedom and Responsibility: Netflix promotes a culture where employees are given the freedom to innovate and the responsibility to meet high expectations. This environment fosters loyalty by trusting employees and aligning their personal goals with the company's success.

ATLASSIAN'S TEAM PLAYBOOK: Atlassian's focus on team health and collaboration is a testament to their commitment to loyalty. The Team Playbook provides strategies and practices to help teams work better together, building a culture of trust and support.

LOYALTY, as a principle of the Bushido Code, is essential for modern DevOps practices. By demonstrating commitment to team and organizational goals, fostering trust, and supporting one another, teams can build a more cohesive and successful environment. As we move forward, integrating loyalty into our daily work can help us build a more dependable and effective software development ecosystem.

CHAPTER 7: EXERCISES

Exercise 1: Team Goal Alignment Workshop

Objective: Align individual and team goals with organizational objectives to foster loyalty and commitment.

INSTRUCTIONS:

1. Organize a workshop where team members can discuss the organization's mission, vision, and goals.
2. Have each team member outline their personal and professional goals.
3. Facilitate a discussion on how individual goals can align with and support the team's and organization's objectives.
4. Create a plan to integrate these aligned goals into daily work and long-term projects.
5. Reflect on how aligning goals fosters loyalty to the team and the organization.

. . .

OUTCOME: Clear alignment of individual, team, and organizational goals, leading to increased commitment and loyalty.

Exercise 2: Trust-Building Activities

Objective: Strengthen trust within the team through activities that encourage collaboration and mutual support.

INSTRUCTIONS:

1. Plan a series of trust-building activities, such as trust falls, team-building games, or collaborative problem-solving tasks.
2. Ensure activities require team members to rely on each other and work together towards a common goal.
3. After each activity, facilitate a discussion on how trust was built and how it can be applied in daily work.
4. Encourage team members to share experiences where trust played a key role in overcoming challenges.
5. Reflect on how these activities enhance trust and loyalty within the team.

OUTCOME: Stronger trust among team members, leading to increased collaboration and loyalty.

Exercise 3: Mentorship Program

Objective: Foster loyalty and support by pairing experienced team members with newer or less experienced colleagues.

INSTRUCTIONS:

1. Establish a mentorship program where experienced team members mentor newer or less experienced colleagues.
2. Pair mentors and mentees based on skills, interests, and development needs.
3. Schedule regular check-ins for mentors and mentees to discuss progress, challenges, and goals.
4. Provide resources and support for mentors to guide their mentees effectively.
5. Reflect on how mentorship builds loyalty by creating supportive relationships and fostering professional growth.

OUTCOME: Enhanced professional development and stronger bonds within the team, leading to increased loyalty.

Exercise 4: Loyalty Recognition Program

Objective: Recognize and reward team members for their loyalty and dedication to the team and organization.

INSTRUCTIONS:

1. Establish a program to recognize and reward team members who demonstrate exceptional loyalty and dedication.

2. Create criteria for recognition, such as years of service, commitment to team goals, or extraordinary support during challenging times.

3. Celebrate recognized individuals during team meetings or through internal communications.

4. Encourage team members to share stories of loyalty and dedication they have witnessed or experienced.

5. Reflect on the impact of recognition on fostering a culture of loyalty and commitment.

OUTCOME: Increased motivation and a culture that values and recognizes loyalty and dedication.

Exercise 5: Reflective Journaling on Loyalty

Objective: Encourage team members to reflect on their personal experiences and commitment to the team and organization.

INSTRUCTIONS:

1. Ask team members to keep a reflective journal where they can write about their experiences related to loyalty and commitment.

2. Provide prompts such as: Describe a time when you felt particularly loyal to the team, How do you show your

commitment to the team's goals? or What actions can you take to strengthen your loyalty to the organization?

3. Schedule regular times for team members to share their reflections in small groups or pairs.

4. Facilitate discussions on how personal reflections can translate into actions that support team and organizational loyalty.

5. Reflect on how these reflections and discussions contribute to a deeper sense of loyalty and commitment.

OUTCOME: Personal insights and a stronger sense of loyalty and commitment within the team.

EIGHT

CHAPTER 8: SELF-CONTROL (自制, JISEI) - MASTERY OVER EMOTIONS AND DESIRES

Discipline refined,
Mastery in every move,
Control brings success.

Self-control, or Jisei, is a vital principle of the Bushido Code, emphasizing mastery over one's emotions and desires. For the samurai, self-control was about maintaining discipline and composure, even in the face of adversity. This principle ensured that their actions were guided by reason and ethics rather than impulsive emotions.

THE ESSENCE of Self-Control in Bushido

Self-control involves:

- Discipline: Adhering to rules and maintaining a rigorous practice to master one's skills.
- Composure: Staying calm and collected, especially in stressful situations.

- Restraint: Controlling impulses and avoiding rash actions.

- Focus: Maintaining concentration on one's goals and duties.

STORIES AND EXAMPLES of Self-Control in Samurai History

One notable example of self-control is the story of Miyamoto Musashi, a legendary samurai and swordsman. Musashi was known for his disciplined training regimen and his ability to remain calm and focused during duels. His mastery over his emotions and desires enabled him to become one of the greatest swordsmen in Japanese history.

APPLYING Self-Control in DevOps

In the context of DevOps, self-control translates to maintaining discipline in processes, ensuring consistency, and staying focused on long-term goals. It involves:

- Practicing Discipline in Processes and Automation: Adhering to established processes and best practices ensures reliability and efficiency. This includes continuous integration and continuous deployment (CI/CD) practices, automated testing, and regular code reviews.

- Maintaining Composure in Crisis: Staying calm and focused during incidents and outages. Effective incident management and postmortem practices are essential for resolving issues and preventing future occurrences.

- Controlling Impulses: Avoiding the temptation to take shortcuts that could compromise quality or security. This includes resisting the urge to push untested code or bypassing critical review processes.

- Staying Focused on Long-Term Goals: Balancing immediate needs with long-term objectives. This involves strategic planning, prioritizing technical debt reduction, and investing in infrastructure improvements.

IMPLEMENTATIONS

Google's Site Reliability Engineering (SRE): Google's SRE teams exemplify self-control through disciplined processes and automation. By adhering to rigorous standards and best practices, they ensure the reliability and scalability of Google's services.

FACEBOOK'S CODE REVIEW PROCESS: Facebook maintains a stringent code review process to ensure code quality and consistency. This disciplined approach helps prevent issues and maintains high standards, reflecting the principle of self-control.

SELF-CONTROL, as a principle of the Bushido Code, is essential for modern DevOps practices. By maintaining discipline in processes, ensuring consistency, and staying focused on long-term goals, teams can build a more reliable and efficient software development environment. As we move forward, integrating self-control into our daily work can help us build a more resilient and effective DevOps culture.

CHAPTER 8: EXERCISES

Exercise 1: Mindfulness Meditation Sessions

Objective: Enhance self-control and emotional regulation through regular mindfulness meditation practices.

INSTRUCTIONS:

1. Schedule regular mindfulness meditation sessions (e.g., weekly or bi-weekly) for the team.
2. Begin each session with a brief explanation of the benefits of mindfulness for self-control and emotional regulation.
3. Guide the team through a 10-15 minute mindfulness meditation practice focused on breathing and awareness.
4. Encourage team members to reflect on their experiences and share how mindfulness helps them maintain self-control in their work.
5. Provide resources for team members to practice mindfulness on their own.

. . .

OUTCOME: Improved self-control and emotional regulation, leading to a more focused and composed team.

Exercise 2: Stress Management Workshops

Objective: Equip team members with techniques to manage stress and maintain composure under pressure.

INSTRUCTIONS:

1. Organize stress management workshops led by a professional or experienced facilitator.
2. Cover techniques such as deep breathing exercises, progressive muscle relaxation, time management, and cognitive-behavioral strategies.
3. Have team members practice these techniques during the workshop.
4. Encourage team members to integrate these stress management practices into their daily routines.
5. Reflect on how managing stress effectively enhances self-control and overall well-being.

OUTCOME: Better stress management skills and improved ability to maintain self-control under pressure.

Exercise 3: Self-Discipline Challenges

Objective: Foster self-discipline through personal and professional challenges that require consistent effort and dedication.

INSTRUCTIONS:

1. Encourage team members to set personal or professional self-discipline challenges, such as learning a new skill, improving a habit, or completing a project.
2. Create a supportive environment where team members can share their goals and progress.
3. Schedule regular check-ins to discuss challenges, successes, and strategies for maintaining discipline.
4. Celebrate the completion of challenges and reflect on the role of self-discipline in achieving success.
5. Document and share success stories to inspire and motivate others.

OUTCOME: Enhanced self-discipline and a culture of continuous improvement and perseverance.

Exercise 4: Incident Response Drills

Objective: Build self-control and composure in high-pressure situations through simulated incident response drills.

INSTRUCTIONS:

1. Plan and execute regular incident response drills that simulate real-world scenarios (e.g., system outages, security breaches).

2. Assign roles and responsibilities to team members and guide them through the incident response process.

3. Focus on maintaining composure, following established protocols, and communicating effectively.

4. Conduct a post-drill review to discuss what went well and areas for improvement.

5. Reflect on how practicing incident response drills enhances self-control and preparedness.

OUTCOME: Improved ability to maintain self-control and composure during actual incidents, leading to more effective incident management.

Exercise 5: Reflection and Goal-Setting Journals

Objective: Encourage self-awareness and self-control through regular reflection and goal-setting practices.

INSTRUCTIONS:

1. Provide team members with journals for regular reflection and goal-setting.

2. Encourage team members to set specific, measurable goals related to self-control, such as managing time better, reducing procrastination, or improving focus.

3. Schedule regular intervals for team members to

reflect on their progress, challenges, and strategies for improvement.

4. Facilitate discussions or one-on-one sessions to share reflections and provide mutual support.

5. Reflect on how regular reflection and goal-setting contribute to personal growth and self-control.

OUTCOME: Increased self-awareness, improved self-control, and a culture of personal and professional growth.

CONCLUSION

The Bushido Code and DevOps work together to make a strong framework for current software development. DevOps values, like working together, always getting better, automating tasks, and being resilient, are very similar to the Bushido Code's virtues, like respect, discipline, and dedication to doing the best. In all parts of work, both methods stress how important honesty, accountability, and respect are.

BY APPLYING the principles of Bushido to DevOps, businesses can create an atmosphere of excellence and moral behavior. This method not only makes things run more smoothly and quickly, but it also makes people feel proud of the work they're doing. When coders and operations teams treat their work with the same respect as samurai, the software they make is new, reliable, and well-liked.

. . .

IT IS my hope that this framework will encourage people and groups to follow these timeless ideals. They can be more successful and make the workplace more creative, honest, and high-performing if they do this. When you combine DevOps and Bushido, you can keep growing and getting better, making sure that the process of making software is just as worthy as the end result.

RECOMMENDED READING : DEVOPS

Foundational Books

- **The Phoenix Project**: A Novel About IT, DevOps, and Helping Your Business Win by Gene Kim, Kevin Behr, and George Spafford
- **The DevOps Handbook**: How to Create World-Class Agility, Reliability, & Security in Technology Organizations by Gene Kim, Patrick Debois, John Willis, and Jez Humble
- **Accelerate**: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations by Nicole Forsgren, Jez Humble, and Gene Kim
- **The Unicorn Project**: A Novel about Developers, Digital Disruption, and Thriving in the Age of Data by Gene Kim
- **Continuous Delivery**: Reliable Software Releases through Build, Test, and Deployment Automation by Jez Humble and David Farley

Advanced and Specialized Books

- **Site Reliability Engineering**: How Google Runs Production Systems by Niall Richard Murphy, Betsy Beyer, Chris Jones, and Jennifer Petoff
- **The Site Reliability Workbook**: Practical Ways to Implement SRE by Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara, and Stephen Thorne
- **Infrastructure as Code**: Managing Servers in the Cloud by Kief Morris
- **Kubernetes**: Up and Running: Dive into the Future of Infrastructure by Kelsey Hightower, Brendan Burns, and Joe Beda
- **Effective DevOps**: Building a Culture of Collaboration, Affinity, and Tooling at Scale by Jennifer Davis and Katherine Daniels

DevOps Practices and Tools

- **Mastering Jenkins** by Jonathan McAllister
- **Terraform**: Up & Running: Writing Infrastructure as Code by Yevgeniy Brikman
- **Docker Deep Dive** by Nigel Poulton
- **Kubernetes Patterns**: Reusable Elements for Designing Cloud-Native Applications by Bilgin Ibryam and Roland Huß

- **Ansible**: Up and Running: Automating Configuration Management and Deployment the Easy Way by Lorin Hochstein and Rene Moser

---

Culture and Leadership

- **Leading the Transformation**: Applying Agile and DevOps Principles at Scale by Gary Gruver and Tommy Mouser
- **Team Topologies**: Organizing Business and Technology Teams for Fast Flow by Matthew Skelton and Manuel Pais
- **The Art of Scalability**: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise by Martin L. Abbott and Michael T. Fisher
- **The Lean Startup**: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses by Eric Ries
- **The Toyota Way**: 14 Management Principles from the World's Greatest Manufacturer by Jeffrey K. Liker

---

Continuous Integration/Continuous Deployment (CI/CD)

- **Continuous Integration**: Improving Software Quality and Reducing Risk by Paul M. Duvall, Steve Matyas, and Andrew Glover
- **DevOps for Developers** by Michael Hüttermann
- **Jenkins 2**: Up and Running: Evolve Your Deployment Pipeline for Next Generation Automation by Brent Laster
- **Pipeline as Code**: Continuous Delivery with Jenkins, Kubernetes, and Terraform by Mohamed Labouardy
- **Agile Application Security**: Enabling Security in a Continuous Delivery Pipeline by Laura Bell, Michael Brunton-Spall, Rich Smith, and Jim Bird

---

Cloud and Microservices

- **Building Microservices**: Designing Fine-Grained Systems by Sam Newman
- **Cloud Native DevOps with Kubernetes**: Building, Deploying, and Scaling Modern Applications in the Cloud by John Arundel and Justin Domingus
- **Microservices Patterns**: With examples in Java by Chris Richardson
- **Designing Data-Intensive Applications**: The Big Ideas Behind Reliable,

Scalable, and Maintainable Systems by Martin Kleppmann
- **Architecting the Cloud**: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS) by Michael J. Kavis

---

Monitoring and Observability

- **Practical Monitoring**: Effective Strategies for the Real World by Mike Julian
- **Prometheus**: Up & Running: Infrastructure and Application Performance Monitoring by Brian Brazil
- **The Art of Monitoring** by James Turnbull
- **Observability Engineering**: Achieving Production Excellence by Charity Majors, Liz Fong-Jones, and George Miranda
- **Monitoring Distributed Systems**: Case Studies from Google, Datadog, and Dropbox by Kevin Crawley

---

Articles, Blogs, and Online Resources

- **DevOps.com** - Comprehensive articles, news, and resources about DevOps.
- **The New Stack** - Articles and analysis on DevOps, cloud native technologies, and software development.

- **InfoQ** - In-depth articles, presentations, and interviews on DevOps, software engineering, and agile.
- **DZone** - DevOps Zone offers a wealth of articles, tutorials, and resources on various DevOps topics.
- **Medium DevOps** - A collection of DevOps stories and articles from practitioners and thought leaders.

Online Courses and Tutorials

- **Coursera**: DevOps and SRE Fundamentals by Google Cloud
- **Udacity**: Cloud DevOps Engineer Nanodegree by Udacity
- **Pluralsight**: DevOps Learning Path - Various courses covering DevOps fundamentals and advanced topics.
- **Linux Academy**: DevOps Playbook - Comprehensive DevOps training and certification.
- **edX**: Introduction to DevOps: Transforming and Improving Operations by The Linux Foundation

---

Conferences and Talks

- **DevOps Enterprise Summit** - Talks and presentations from leaders and practitioners in DevOps.
- **KubeCon + CloudNativeCon** - Conferences focused on Kubernetes and cloud-native technologies.
- **All Day DevOps** - A free, online DevOps conference featuring global experts.
- **SREcon** - A conference dedicated to site reliability engineering and DevOps practices.
- **DockerCon** - Sessions and workshops on containerization, Docker, and DevOps.

JARGON : DEVOPS

A

Agile: A methodology for software development that emphasizes incremental delivery, collaboration, continual planning, and continual learning.

ARTIFACT: Any by-product produced during the development of software. Examples include compiled code, build logs, and other data.

B

Blue-Green Deployment: A release management strategy that reduces downtime and risk by running two identical production environments. Only one environment (blue or green) serves live production traffic.

. . .

BUILD: The process of converting source code files into standalone software artifacts that can be run on a computer.

C

CI/CD (Continuous Integration/Continuous Deployment): A set of practices that aim to integrate code changes frequently, verify them through automated testing, and deploy them into production automatically.

CONFIGURATION MANAGEMENT: The practice of handling changes systematically so that a system maintains its integrity over time. Tools like Puppet, Chef, and Ansible are commonly used.

CONTAINERIZATION: The process of packaging an application and its dependencies together in a container to ensure it runs consistently across different computing environments. Docker is a popular containerization platform.

CANARY RELEASE: A deployment strategy to release a new version of an application to a small subset of users to test in the production environment before rolling it out to the entire user base.

D

DevOps: A set of practices that combine software development (Dev) and IT operations (Ops) aimed at shortening the

development lifecycle and providing continuous delivery with high software quality.

DEPLOYMENT: The process of moving an application from a development environment to a production environment where it can be used by end-users.

DOWNTIME: A period when a system is not operational or accessible.

E

Elasticity: The ability of a system to dynamically increase or decrease resources as needed.

ENVIRONMENT: A designated space where specific aspects of the software development lifecycle occur, such as development, testing, staging, and production environments.

F

Feature Toggle: A technique to enable or disable features in a software application without deploying new code. It allows new features to be tested in production and rolled back if necessary.

FEEDBACK LOOP: A process where outputs of a system are circled back and used as inputs. In DevOps, this often

refers to the continuous feedback from monitoring and logging systems used to improve software and processes.

G

Git: A distributed version control system used for tracking changes in source code during software development.

GITOPS: An operational framework that uses Git as a single source of truth for declarative infrastructure and applications.

H

High Availability (HA): A system design approach that ensures a certain degree of operational continuity during a given measurement period.

HYBRID CLOUD: A computing environment that combines on-premises, private cloud, and public cloud services with orchestration between the platforms.

I

Immutable Infrastructure: An approach where servers or components are not modified after they are deployed. Any changes result in new servers or components being built.

INFRASTRUCTURE AS CODE (IAC): The process of managing and provisioning computing infrastructure

through machine-readable configuration files, rather than through physical hardware configuration or interactive configuration tools.

J

Jenkins: An open-source automation server that helps automate parts of the software development process related to building, testing, and deploying.

JIRA: A tool developed by Atlassian used for bug tracking, issue tracking, and project management.

K

Kubernetes: An open-source container orchestration platform designed to automate deploying, scaling, and operating containerized applications.

L

Log Management: The practice of dealing with large volumes of log data generated by various systems and applications, including their collection, storage, and analysis.

LOAD BALANCER: A device or process that distributes network or application traffic across multiple servers to ensure no single server becomes overwhelmed, thereby improving responsiveness and availability.

M

Microservices: An architectural style that structures an application as a collection of small, autonomous services modeled around a business domain.

MONITORING: The process of collecting, analyzing, and using data to track the performance, availability, and overall health of applications and infrastructure.

N

Node: In computing, a node is a single machine or server in a cluster.

O

Orchestration: The automated configuration, management, and coordination of computer systems, applications, and services.

OBSERVABILITY: A measure of how well you can understand the internal state of a system based on the data it produces, such as logs, metrics, and traces.

P

Pipeline: A set of automated processes that enable developers and DevOps professionals to compile, build, and deploy their code to production environments.

. . .

PLAYBOOK: In configuration management, a playbook is a file containing scripts, written in YAML, that automate tasks. Often associated with Ansible.

Q

Quality Assurance (QA): The systematic process of checking to see whether a product or service being developed is meeting specified requirements.

R

Rollback: The process of reverting a system or application to a previous state after a failed or problematic update.

REPOSITORY (REPO): A central location where data is stored and managed. In Git, a repository is used to store code, track changes, and collaborate with others.

S

Scripting: Writing small programs to automate repetitive tasks.

SCRUM: An agile process framework for managing complex knowledge work, with an emphasis on software development.

SERVICE LEVEL AGREEMENT (SLA): A contract

between a service provider and a customer that specifies the level of service expected during its term.

SERVERLESS: A cloud-computing execution model where the cloud provider dynamically manages the allocation and provisioning of servers.

STAGING ENVIRONMENT: A testing environment that is a replica of the production environment where new versions of software are tested before being deployed to production.

T

Terraform: An open-source IaC tool that allows users to define and provide data center infrastructure using a declarative configuration language.

TEST-DRIVEN DEVELOPMENT (TDD): A software development process where tests are written before the code they are meant to validate.

U

Uptime: The amount of time a system or service is operational and accessible.

V

Version Control: The practice of managing and tracking changes to software code, allowing multiple developers to collaborate and maintain a history of changes.

VIRTUAL MACHINE (VM): An emulation of a computer system that provides the functionality of a physical computer.

W

Workflow: A sequence of tasks that processes a set of data. In DevOps, workflows are automated sequences of tasks to build, test, and deploy applications.

X

XP (Extreme Programming): An agile software development framework that aims to produce higher quality software and higher quality of life for the development team.

Y

YAML (YAML Ain't Markup Language): A human-readable data serialization standard that is commonly used for configuration files and in applications where data is being stored or transmitted.

Z

Zero Downtime Deployment: A deployment method that allows an application to be updated without any downtime.

RECOMMENDED READING : BUSHIDO CODE

Foundational Books on Bushido

- **Bushido**: The Soul of Japan by Inazo Nitobe - A classic work that introduces the concept of Bushido to the Western world, exploring the ethical code and philosophy of the samurai.
- **Hagakure**: The Book of the Samurai by Yamamoto Tsunetomo - A collection of thoughts and reflections by a samurai retainer, offering insights into the mindset and values of the samurai class.
- **The Book of Five Rings** by Miyamoto Musashi - A renowned text on strategy, tactics, and philosophy written by the legendary swordsman Miyamoto Musashi, which also reflects Bushido principles.
- **Code of the Samurai**: A Modern Translation of the Bushido Shoshinshu of Taira Shigesuke translated by Thomas Cleary - A

modern translation of a key Bushido text that provides practical advice for living according to samurai principles.
- **The Unfettered Mind**: Writings from a Zen Master to a Master Swordsman by Takuan Soho - Philosophical essays by a Zen monk that influenced many samurai, emphasizing mental discipline and clarity.

Historical and Cultural Context

- **Samurai**: An Illustrated History by Mitsuo Kure - A richly illustrated overview of the samurai's history, culture, and the Bushido code.
- **The Samurai**: A Military History by Stephen Turnbull - A comprehensive history of the samurai, detailing their battles, culture, and the evolution of Bushido.
- **The Taming of the Samurai**: Honorific Individualism and the Making of Modern Japan by Eiko Ikegami - A scholarly examination of the role of the samurai and Bushido in the social and political development of Japan.
- **Warriors of Japan**: As Portrayed in the War Tales by Paul Varley - An analysis of samurai literature, providing insights into the values and ideals of the samurai class.
- **The Secret Traditions of the Shinobi**: Hattori Hanzo's Shinobi Hiden and Other Ninja Scrolls by Antony Cummins and Yoshie

Minami - A look into the world of the ninja and their relationship with the samurai, offering a different perspective on Bushido.

Philosophy and Ethics

- **The Art of Peace** by Morihei Ueshiba - Teachings and philosophy of the founder of Aikido, emphasizing harmony and ethical conduct, reflecting Bushido principles.
- **Zen and the Art of Archery** by Eugen Herrigel - An exploration of Zen practice and its influence on the samurai, focusing on mental discipline and mastery.
- **The Zen Way to Martial Arts**: A Japanese Master Reveals the Secrets of the Samurai by Taisen Deshimaru - Insights from a Zen master on the spiritual and philosophical dimensions of martial arts and Bushido.
- **Soul of the Samurai**: Modern Translations of Three Classic Works of Zen & Bushido by Thomas Cleary - Modern translations of classic Bushido texts, including writings by Yagyu Munenori and Takuan Soho.
- **Budo Secrets**: Teachings of the Martial Arts Masters by John Stevens - A collection of teachings from various martial arts masters, highlighting the ethical and spiritual aspects of Bushido.

Contemporary Reflections on Bushido

- **Living with the Samurai Code**: The Eight Essential Bushido Virtues by Mark Edward Cody - A modern interpretation of Bushido, applying its principles to contemporary life.
- **The Warrior Ethos** by Steven Pressfield - An exploration of the warrior code across cultures, including the samurai, with reflections on modern applications.
- **Training the Samurai Mind**: A Bushido Sourcebook translated by Thomas Cleary - A collection of writings by samurai warriors and scholars, providing a deeper understanding of Bushido philosophy.
- **The Way of the Living Sword**: The Secret Teachings of Yagyu Munenori translated by D.E. Tarver - Translations of teachings by a renowned samurai, focusing on the philosophical and strategic aspects of Bushido.
- **The Samurai Mind**: Lessons from Japan's Master Warriors edited by Christopher Hellman - A compilation of samurai writings that offer insights into the mental discipline and ethical principles of Bushido.

Articles, Blogs, and Online Resources

- **The Art of Manliness**: The Bushido Code: The Eight Virtues of the Samurai - A detailed article exploring the virtues of Bushido and their relevance today.
- **Tofugu: Bushido**: The Samurai Code of Honor - An engaging blog post discussing the history and principles of Bushido.
- **Koryu.com**: Articles on Samurai and Bushido - A collection of articles and essays on traditional martial arts and the philosophy of Bushido.
- **Stanford Encyclopedia of Philosophy**: Bushido - An academic article providing an overview of Bushido and its philosophical implications.

JARGON : BUSHIDO

A

Aiki (合気): The harmonious integration of energy or spirit. In martial arts, it refers to blending with an opponent's movements to neutralize their attack.

ASHI (足): Foot or leg. Often used in martial arts to describe footwork and positioning.

B

Budo (武道): The martial way or the way of war. Refers to the modern practice of martial arts that focuses on personal development and spiritual growth.

BUSHIDO (武士道): The way of the warrior. The ethical code and philosophy that guided the samurai class in feudal

Japan, emphasizing virtues such as loyalty, honor, and discipline.

C

Chugo (忠義): Loyalty and devotion. A central virtue in Bushido, referring to the samurai's unwavering loyalty to their lord and duty.

CHIKARA (力): Strength or power. Important in both physical combat and inner spiritual strength.

D

Do (道): The way or path. Used in the names of various Japanese arts and disciplines, such as Judo (the way of gentleness) and Kendo (the way of the sword).

DOJO (道場): A training hall or place where martial arts are practiced.

E

Eiji (栄治): Glory and honor. Represents the achievements and recognition a samurai strives for through their actions and conduct.

F

Fudoshin (不動心): The immovable mind or steadfast

spirit. A state of mental stability and composure, essential for maintaining focus and clarity under pressure.

FUKU (福): Fortune or luck. Though not a core principle of Bushido, fortune plays a role in the samurai's life and destiny.

G

Gi (義): Righteousness or justice. A fundamental virtue in Bushido, referring to the moral and ethical conduct expected of a samurai.

GIRI (義理): Duty or obligation. The sense of responsibility and duty to fulfill one's role and obligations honorably.

H

Hara (腹): Belly or abdomen. In martial arts, it refers to the center of energy (hara) in the body, often associated with physical and spiritual strength.

HARA-KIRI (腹切り): Ritual suicide by disembowelment, also known as seppuku. Performed by samurai to restore honor after dishonor or failure.

HIKAGE (日影): Shadow or shade. Symbolic of the

hidden aspects of a samurai's life, including humility and modesty.

HONOR (名誉, Meiyo): Upholding reputation and moral integrity. A central tenet of Bushido, reflecting the importance of maintaining one's honor through righteous actions.

I

Iaido (居合道): The way of drawing the sword. A martial art focused on the smooth, controlled drawing and sheathing of a sword.

IKIGAI (生き甲斐): A reason for being. The sense of purpose and motivation that drives a samurai to live a meaningful life.

J

Jin (仁): Benevolence or compassion. A key virtue in Bushido, emphasizing kindness, empathy, and compassion towards others.

JUTSU (術): Technique or skill. Refers to the practical application of martial arts techniques.

K

Kamikaze (神風): Divine wind. Historically, it refers to the

typhoons that protected Japan from Mongol invasions. In World War II, it referred to suicide pilots.

KATANA (刀): The traditional Japanese sword used by samurai. Known for its sharpness and craftsmanship.

KENDO (剣道): The way of the sword. A modern Japanese martial art that uses bamboo swords and protective armor.

KI (気): Vital energy or life force. Central to many Japanese martial arts and spiritual practices.

KIAI (気合): A shout or yell used in martial arts to focus energy and intimidate opponents.

KOKORO (心): Heart or spirit. Represents the inner essence and character of a person.

M

Metsuke (目付): Gaze or focus. In martial arts, it refers to the proper use of sight and awareness in combat.

MUSHIN (無心): No mind or empty mind. A mental state of clarity and focus, free from distractions and emotions.

N

Naginata (薙刀): A pole weapon with a curved blade, used by samurai and warrior monks.

NINJO (人情): Human emotion or compassion. Balances the strictness of duty and obligation with empathy and humaneness.

R

Rei (礼): Politeness or respect. A fundamental aspect of Bushido, reflecting the importance of courteous behavior and respect for others.

RONIN (浪人): A wandering samurai without a master. Often depicted in literature and film as seeking redemption or purpose.

S

Satori (悟り): Enlightenment or awakening. A profound spiritual realization in Zen Buddhism.

SEPPUKU (切腹): Ritual suicide by disembowelment, also known as hara-kiri. A way for samurai to restore honor after disgrace.

. . .

SHOGUN (将軍): The military dictator of Japan during the feudal period, commanding the samurai class.

SHOSHIN (初心): Beginner's mind. An attitude of openness and eagerness, free from preconceptions.

SAMURAI (侍): The warrior class of feudal Japan, known for their adherence to the Bushido code.

SENSEI (先生): Teacher or master. A term of respect for instructors in martial arts and other disciplines.

T

Tatami (畳): Traditional Japanese flooring made of woven straw. Often used in dojos and homes.

TESSEN (鉄扇): Iron fan used by samurai, both as a weapon and a signaling device.

W

Wabi-sabi (侘寂): A Japanese aesthetic that finds beauty in imperfection and impermanence.

Y

Yamato-damashii (大和魂): The Japanese spirit or soul.

Represents the cultural and spiritual essence of the Japanese people.

YARI (槍): A traditional Japanese spear used by samurai in battle.

YOJIMBO (用心棒): Bodyguard or protector. Often depicted in samurai films and literature.

Z

Zen (禅): A school of Mahayana Buddhism that emphasizes meditation and intuition rather than ritual worship or study of scriptures.

ZANSHIN (残心): Remaining mind or awareness. A state of continued alertness and focus after an action is completed.

HAIKUS

Chapter 1: Rectitude (義, Gi)

Code pure and steadfast,
Integrity in each line,
Righteous path we walk.

Chapter 2: Courage (勇, Yū)

Change embraced with strength,
Innovation lights the way,
Fearless hearts prevail.

Chapter 3: Benevolence (仁, Jin)

Kindness in our code,

Teamwork builds a better world,
Compassion in sync.

Chapter 4: Respect (礼, Rei)

Respect guides our hands,
Politeness in every task,
Harmony in work.

Chapter 5: Honesty (誠, Makoto)

Truth in every word,
Open feedback shapes our path,
Honesty prevails.

Chapter 6: Honor (名誉, Meiyo)

Honor in our craft,
Excellence we strive to reach,
Reputation earned.

Chapter 7: Loyalty (忠義, Chūgi)

Loyal to the team,

Commitment through thick and thin,
Trust and faith endure.

Chapter 8: Self-Control (自制, Jisei)

Discipline refined,
Mastery in every move,
Control brings success.