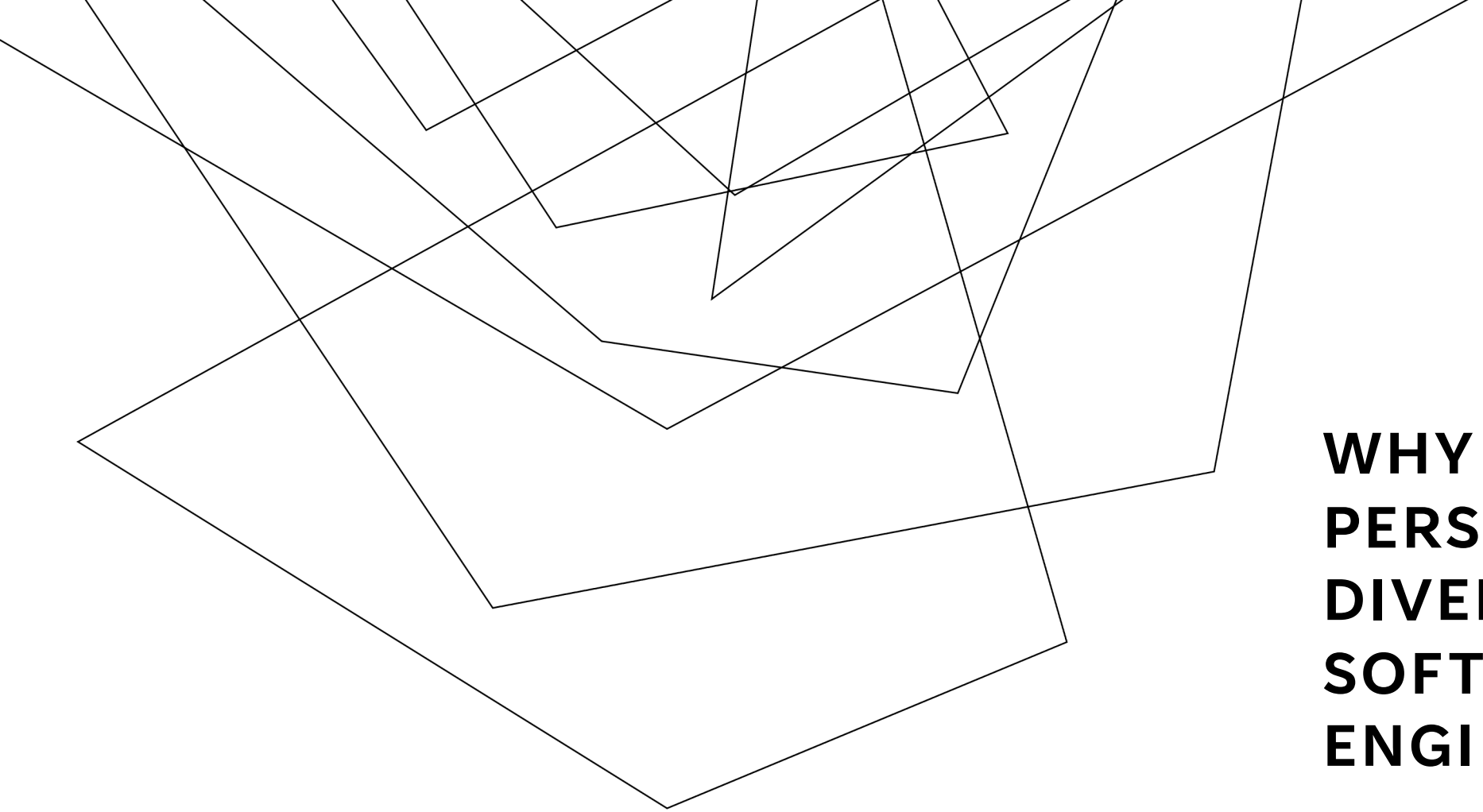


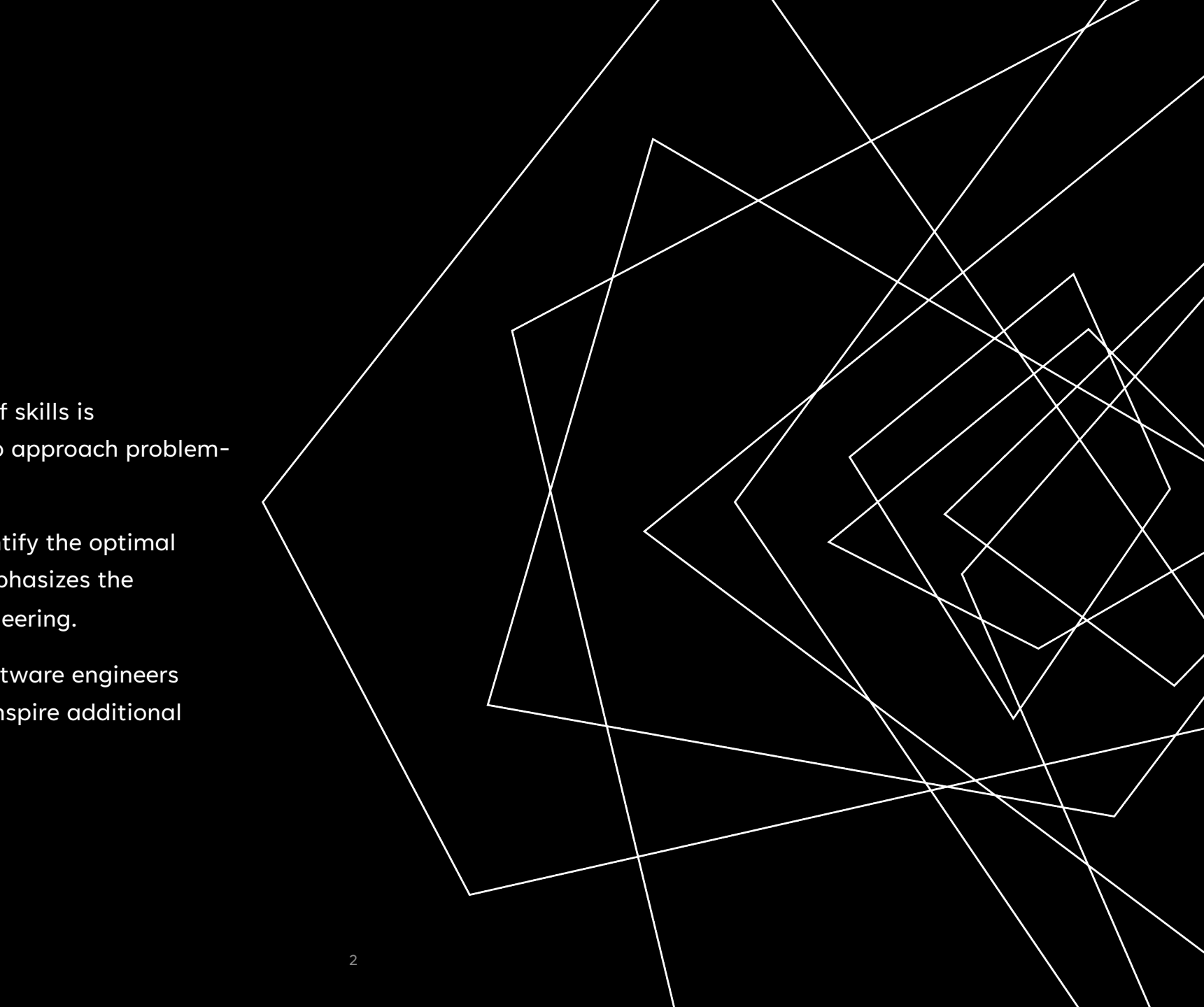
WHY DO WE NEED PERSONALITY DIVERSITY IN SOFTWARE ENGINEERING?

Authors : Luiz Fernando Capretz, Faheem Ahmed
Published in March 2010

-Pavan Kumar Mudven

ABSTRACT

- In software engineering, having a diverse set of skills is advantageous because it enables individuals to approach problem-solving from different angles.
- This study employs psychological types to identify the optimal match for software development tasks and emphasizes the significance of human factors in software engineering.
- The article aims to raise awareness among software engineers regarding the benefits of skill diversity and to inspire additional research in software psychology.



INTRODUCTION

PERSONALITY

The unique combination of traits, behaviors, and thought patterns that make up an individual's character and influence their reactions to the world around them.

DIVERSITY

The recognition and appreciation of differences among individuals, including variations in race, ethnicity, gender, culture, socioeconomic status, and personality.

PERSONALITY DIVERSITY

The range of differences in personality traits, such as openness, agreeableness, extroversion, neuroticism, and conscientiousness, that exist among people.

INCLUSION

The creation of an environment that embraces diversity, fosters a sense of belonging, and values the contributions of all individuals, including those with different personality traits.

IMPORTANCE OF PERSONALITY DIVERSITY

RISK MANAGEMENT

Personality diversity can also play a critical role in managing risks in software engineering. A diverse team can identify and address potential risks and issues from a range of perspectives, leading to better risk management strategies.

INNOVATION

Personality diversity can contribute to innovation in software engineering by enabling team members to approach problems from different perspectives and bring a variety of ideas and solutions to the table.

PERFORMANCE

Personality diversity can enhance the performance of software engineering teams by increasing the overall quality of work, reducing errors, and promoting a more efficient and effective development process.

COLLABORATION

Effective collaboration in software engineering requires a diversity of skills, as team members with different backgrounds and abilities can complement each other and work together to achieve shared goals.

UNDERSTANDING PSYCHOLOGICAL FACTORS IN SOFTWARE ENGINEERING

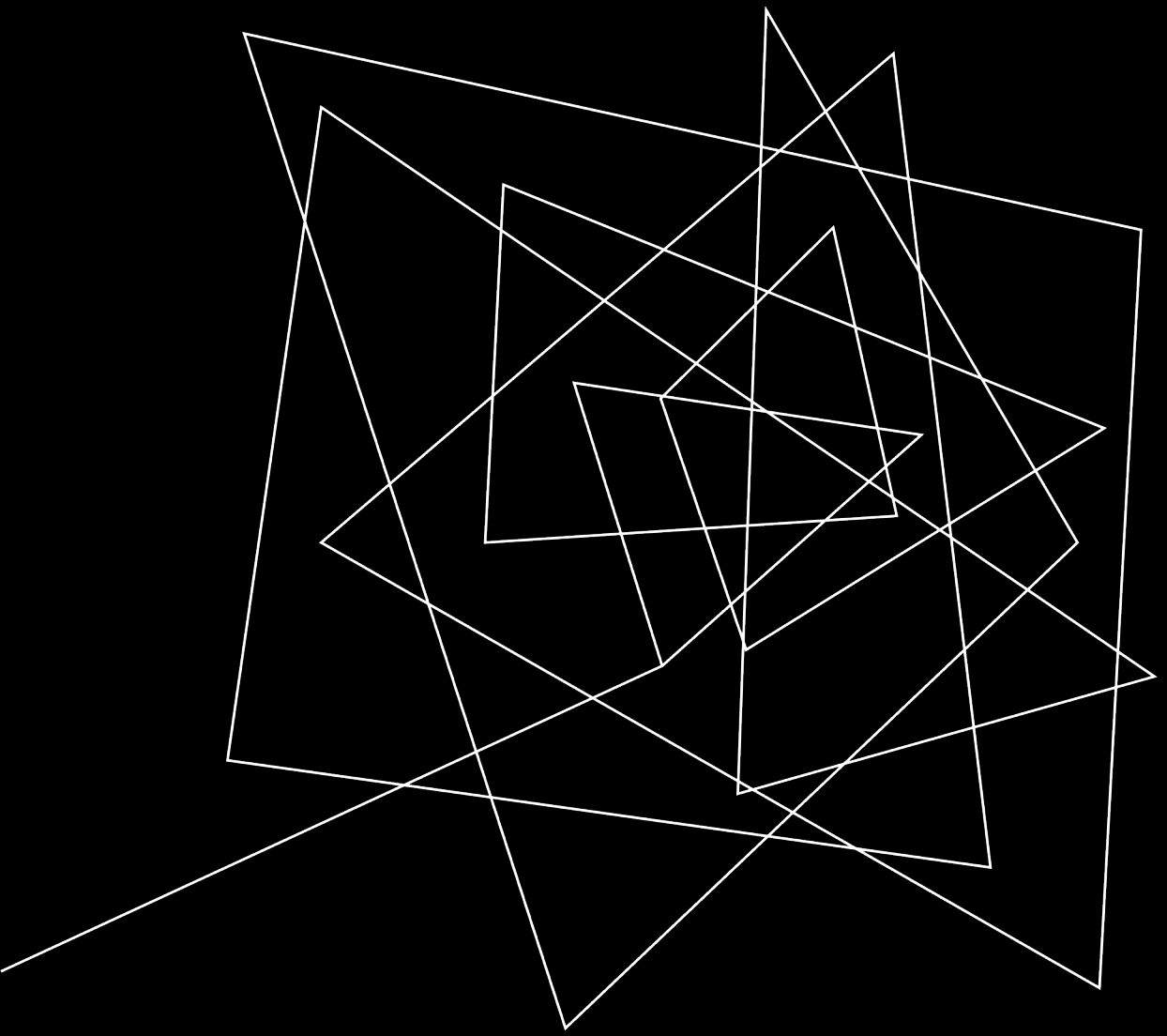
Psychological factors in software engineering refer to the cognitive and emotional processes that influence the behavior of individuals and teams involved in the development and maintenance of software systems.

These factors can include:

- **Perception and cognition**: How developers perceive and interpret information, solve problems, make decisions, and reason about the software system.
- **Emotions and motivation**: How developers feel about their work, the project, and their colleagues, and what drives their behavior and productivity.
- **Personality and individual differences**: How individual differences in personality traits, values, and attitudes affect developers' communication, collaboration, and performance.
- **Team dynamics**: How interpersonal relationships, communication patterns, and leadership styles within software development teams affect productivity, creativity, and innovation.
- **Stress and burnout**: How workload demands, project complexity, and job satisfaction affect developers' mental health and well-being, and ultimately their performance and productivity.

BENEFITS OF USING PSYCHOLOGICAL FACTORS IN SOFTWARE ENGINEERING

- **Understanding user behavior**: Psychological factors such as cognitive and behavioral psychology can help software engineers understand how users interact with software systems. This knowledge can be used to design more user-friendly interfaces and improve the overall user experience.
- **Motivating team members**: Psychology can also help managers motivate their team members by understanding what motivates them and tailoring their management style accordingly. For example, some team members may be motivated by praise and recognition, while others may be motivated by more challenging projects or opportunities for growth.
- **Improving collaboration**: Psychology can help improve collaboration and communication among team members. By understanding the different communication styles of team members and addressing any conflicts that arise, software engineers can work more effectively together.
- **Reducing errors**: Psychological factors such as attention and memory can be used to improve software quality by reducing errors. For example, software engineers can design software systems that consider the limits of human attention and memory, which can help prevent errors and improve the reliability of the software.



INVESTIGATING THE HUMAN FACTORS IN SOFTWARE ENGINEERING

VARIOUS TYPES OF HUMAN FACTORS

INDIVIDUAL FACTOR

This type of factor specifically refers to the **software developers** themselves:

Social and psychosocial issues: they can include (Financial Issues, Family Issues, Health Issues,... etc.) which have an impact on the behavior of the programmer or software engineer.

Technical skills weaknesses: when the programmer has no enough experience in programming which may reflects on the efficiency of his outputs

GROUP FACTOR

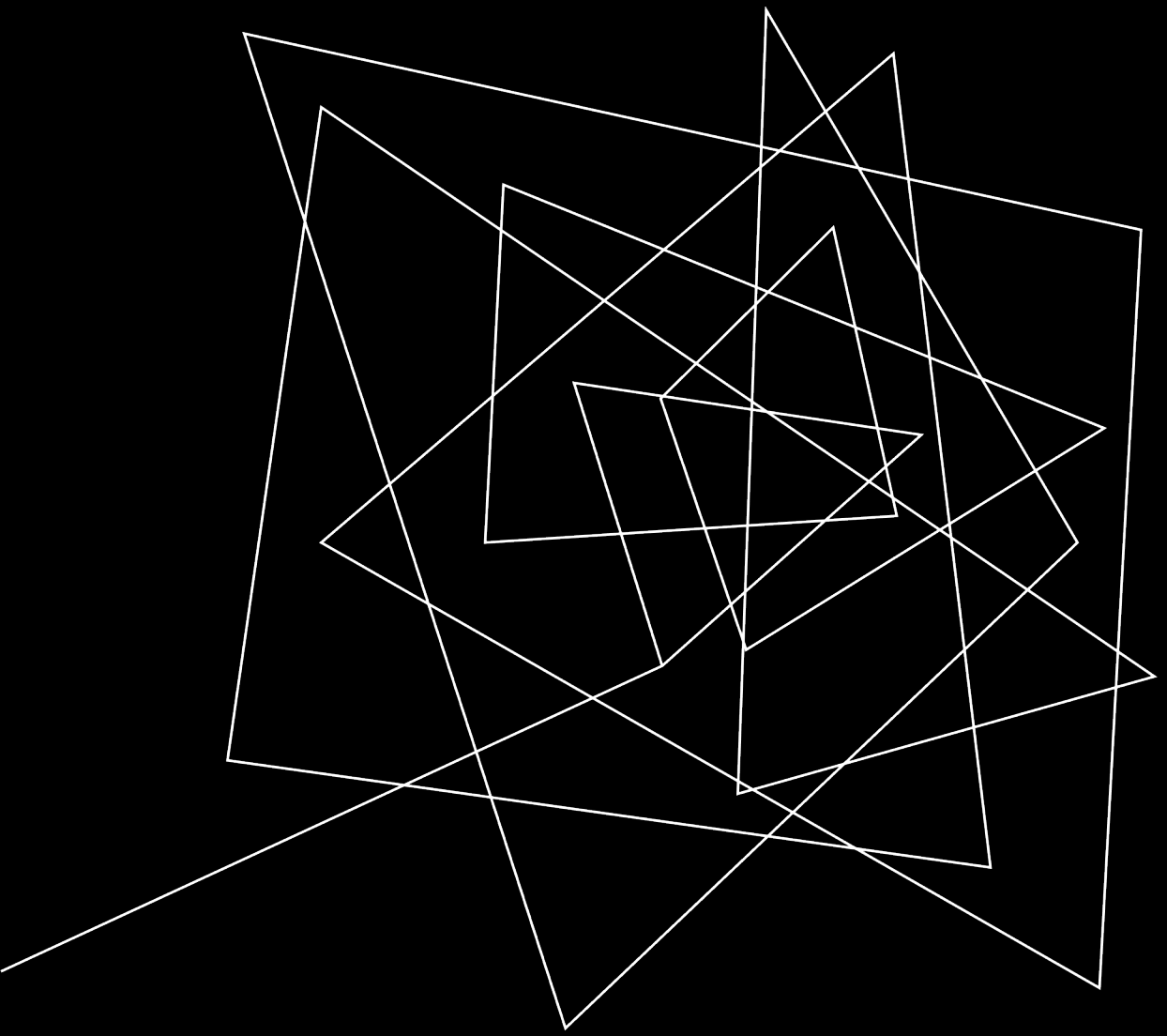
This class of factors refers to a failure in the software engineering team, such as a failure to grasp variable names due to cultural differences, for instance (the color white in French called "blanc" which is like the pronunciation of "blue" in English). This kind of miscommunication among team members may lead to serious issues.

EXTERNAL FACTOR

There are several external human factors that might have an impact on software engineering. Some of these issues are brought on by governmental laws or directives that have a bad impact on the field. The social and technological debt is the second example of external human forces.

DEALING WITH HUMAN FACTORS

- **Make a following up with programmers to check there social and psychosocial situation.**
- **Make a list of a minimum skills needed to hire programmers corresponds with the needs of the software engineering projects.**
- **Take in consideration the culture background in the phase of composing a software engineering team.**
- **In the side of country strategies, the implementation of policies that enhance software engineering in state policies must be considered.**



FINDINGS & CONCLUSION

MYERS-BRIGGS TYPE INDICATOR (MBTI)

- It is a personality assessment tool that measures and categorizes individuals based on their preferences for how they perceive and interact with the world around them. It identifies 16 different personality types based on four main parameters.
- The MBTI is a popular tool used in the school, workplace and personality development programs.
- The MBTI has four parameters for assessing personality types.
- Everyone prefers some personality qualities over others, as indicated by their scores on the MBTI scales.

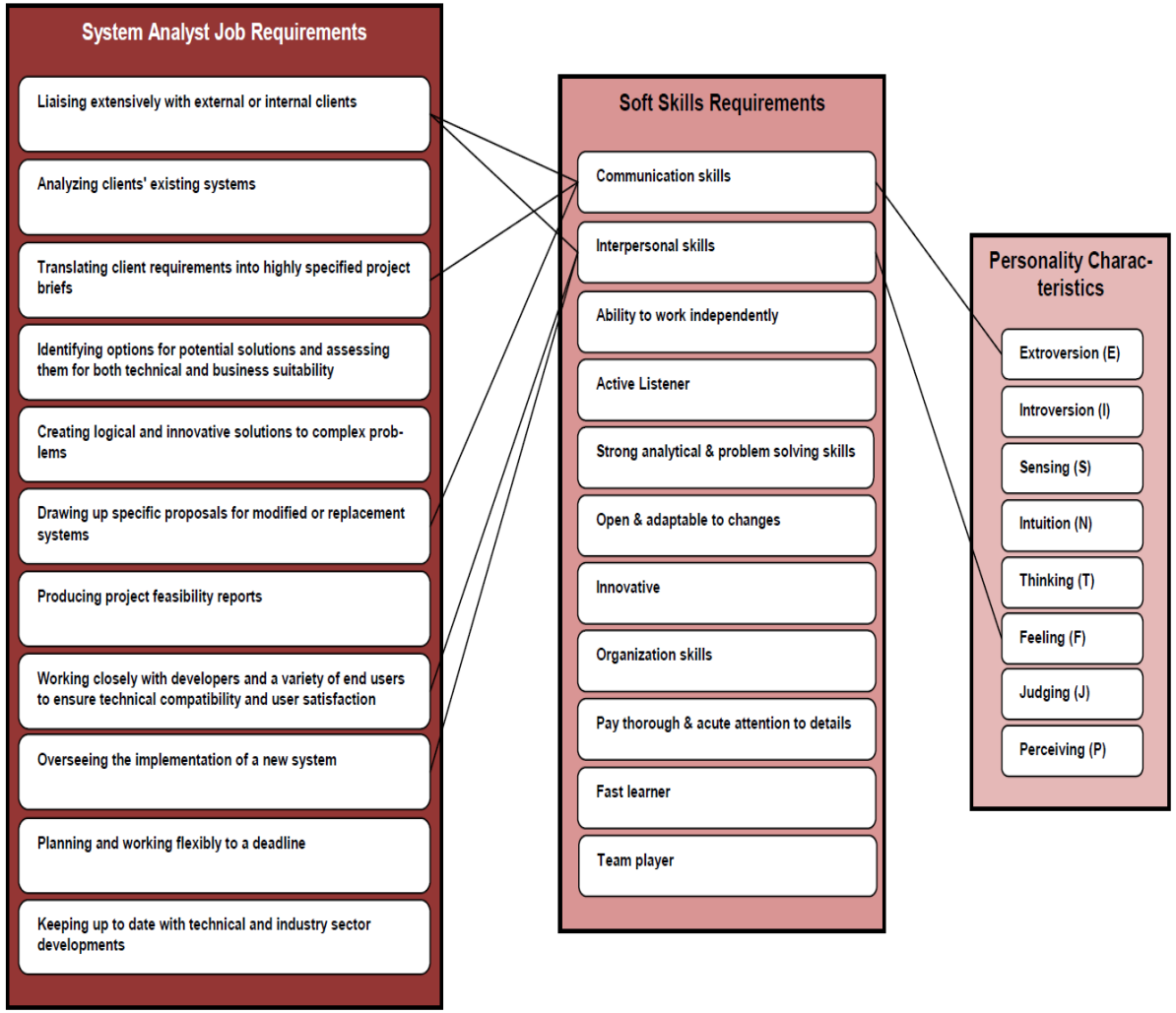
The MBTI establishes four parameters for assessing personality types, which are:

- Extroversion (E) - Introversion (I): This scale measures whether individuals prefer to focus their attention on the outer world of people and things (Extraversion) or on their inner world of ideas and concepts (Introversion).
- Sensing (S) - Intuition (N): This scale measures how individuals prefer to gather information. Sensing types prefer to focus on facts and details, while Intuitive types prefer to focus on the big picture and possibilities.
- Thinking (T) - Feeling (F): This scale measures how individuals prefer to make decisions. Thinking types prioritize logical analysis and objectivity, while Feeling types prioritize personal values and empathy.
- Judging (J) - Perceiving (P): This scale measures how individuals prefer to organize their lives. Judging types like to have a plan and stick to it, while Perceiving types prefer to keep their options open and adapt to changing circumstances.

FINDINGS

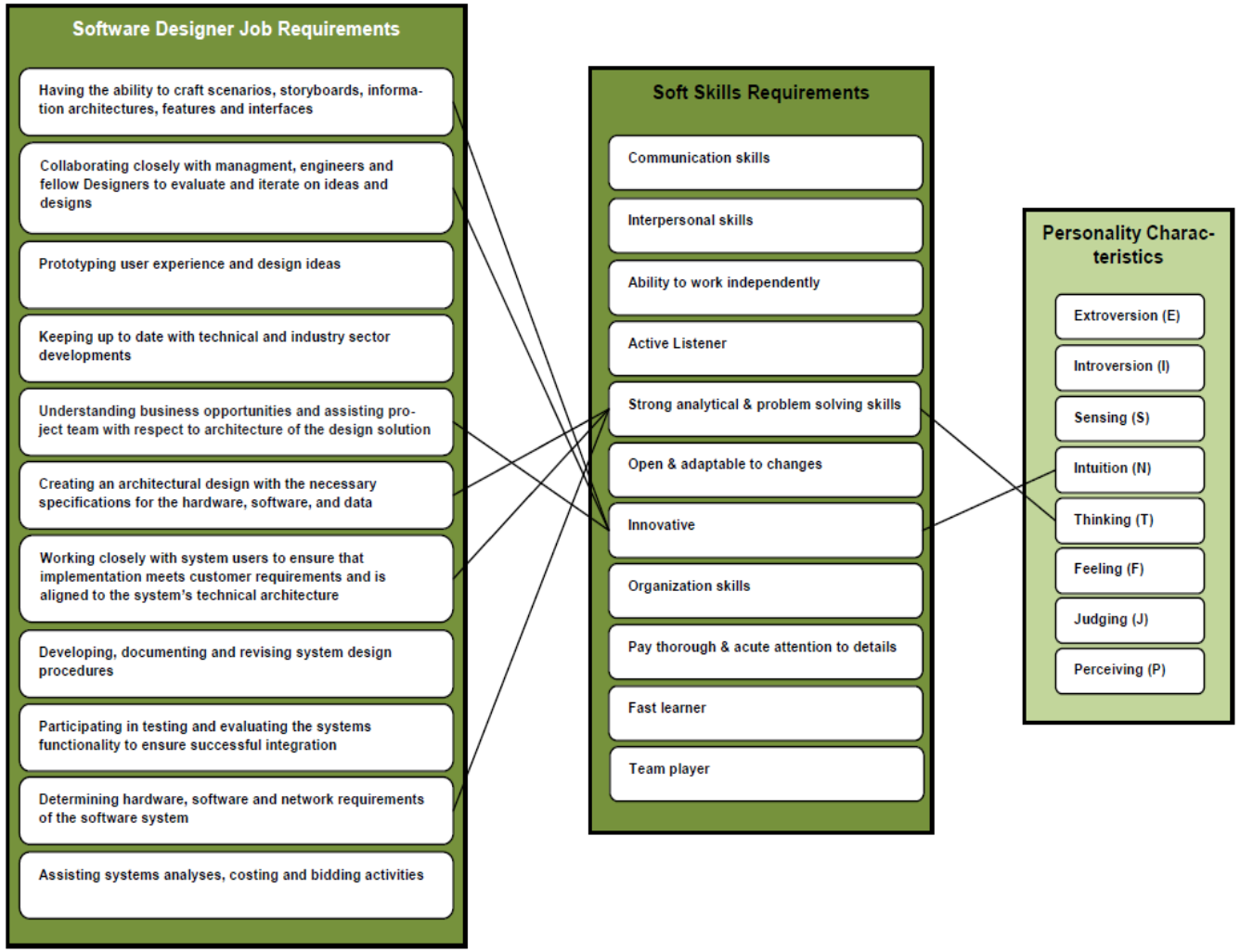
SYSTEM ANALYSIS

- System analysis phase involves identifying high-level components of a real-world application and creating an abstract model of the application based on user and client needs and requirements.
- System analysts need to have strong communication and interpersonal skills to interact with users and clients effectively. Extroversion (E) and feeling (F) personality traits are particularly desirable for this role.
- System analysts must possess strong analytical and problem-solving skills, be able to work independently, and be adaptable to changes. They also need to be fast learners and team players.
- The main job requirements of system analysts include liaising with clients, analyzing existing systems, creating project briefs, identifying potential solutions, producing feasibility reports, overseeing implementation, and keeping up to date with industry developments.



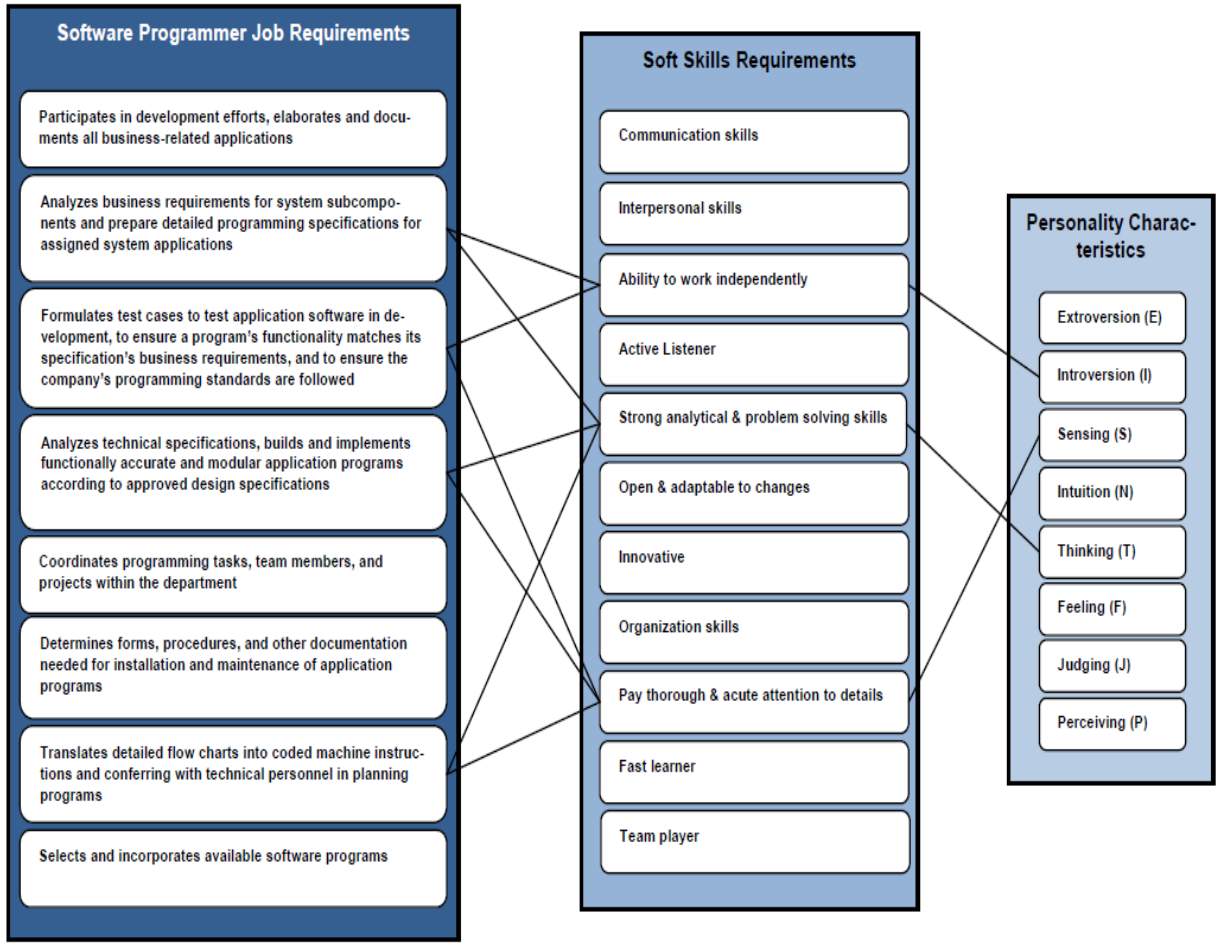
SOFTWARE DESIGN

- Software designers should have the ability to see the big picture, isolate relevant information from large quantities of data, and discern patterns. They should be intuitive, imaginative, and innovative.
- Software designers perform tasks such as prototyping, elaborating processing functions, and defining inputs and outputs. The first part of the design stage may require characteristics like those needed for analysis, including team discussions and interaction with the user.
- Intuition (N) and thinking (T) characteristics are highly desirable for software designers, whereas feeling (F) is only somewhat desirable. Designers should be able to predict how users will feel about the design.
- A combination of judging (J) and perceiving (P) characteristics would ensure that the best design solution is found, rather than simply the first one.



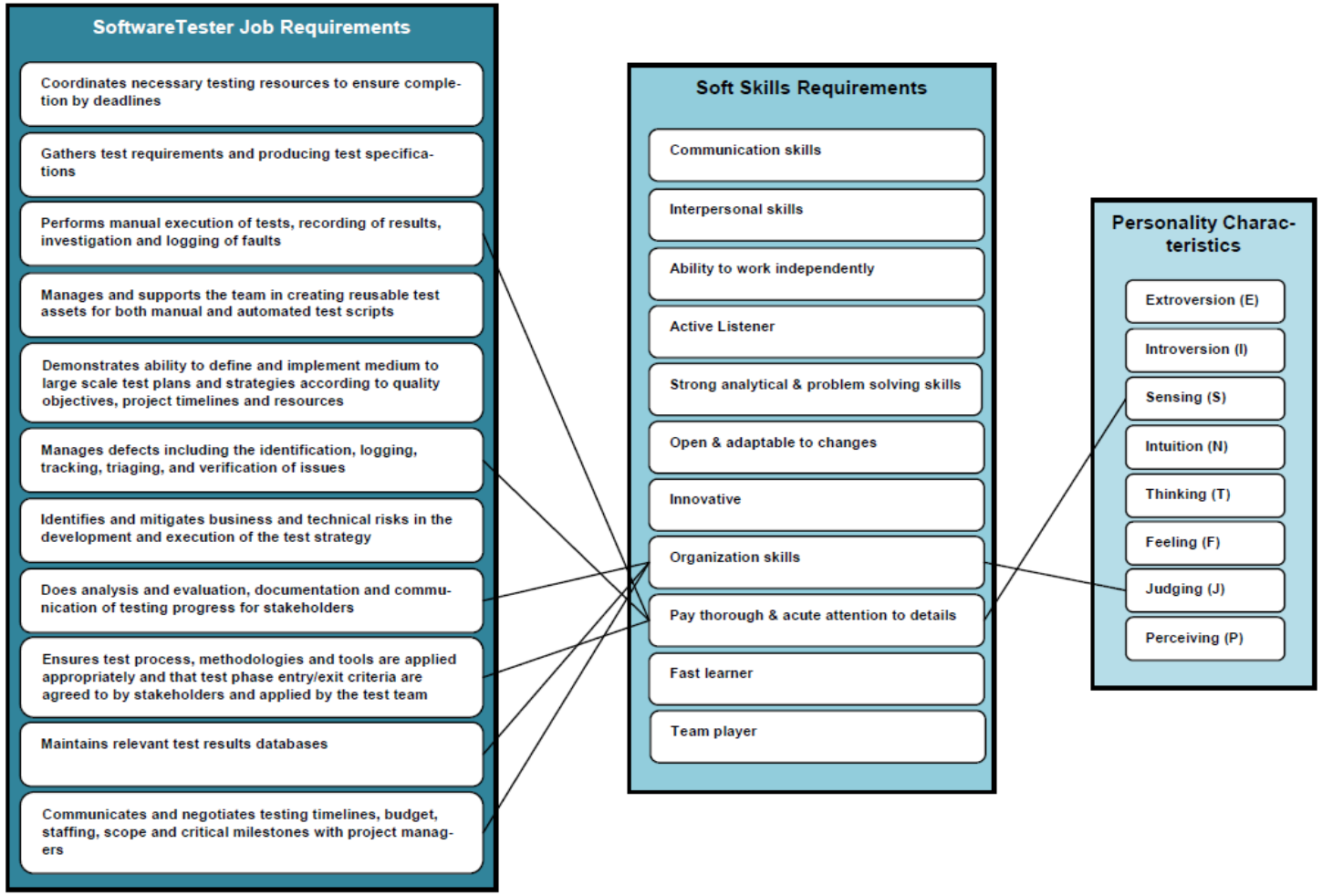
PROGRAMMER

- Programming involves translating a refined version of the design into programs through an iterative stepwise refinement process, requiring attention to detail and logical, analytical thinking.
- The programming stage is more suitable for thinkers (T) than feelers due to the problem of interpreting and giving meaning to variables.
- Programming tasks demand little interpersonal contact, making it an essentially solitary activity.
- Programmers working with the specifications from designers need to be logical (Thinkers), pay attention to details (Sensing), and have the capacity to work independently (Introverts).



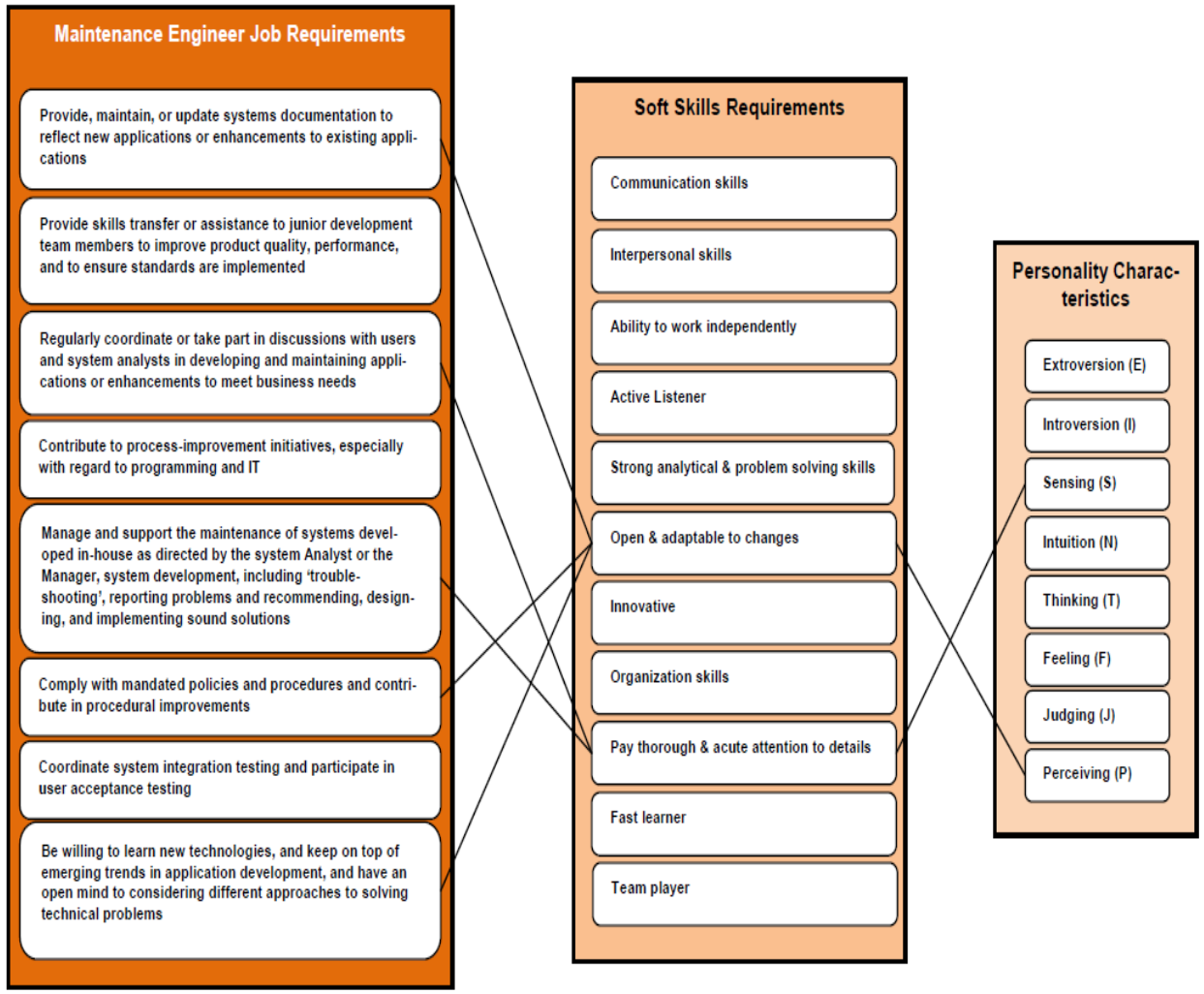
TESTING

- Testing is the process of finding defects in software, and it is not limited to the testing phase alone. Defects can emerge in earlier stages such as system analysis and design phases.
- Testing techniques include unit testing, integration testing, and system testing. Unit testing involves testing individual modules to ensure they function properly with expected and unexpected inputs based on the module's specification. Integration testing verifies well-defined interfaces among collections of modules. System testing is the process of validating the entire software system.
- Testing strategies require a methodical and systematic approach, and debugging can be emotionally challenging for software engineers.
- Testing demands attention to detail and is often performed under pressure to meet deadlines. Individuals with precision (Sensing) and order (Judging) characteristics are highly desirable for testing, as the process requires persistence, choosing from a wide range of possibilities, and paying incredible attention to detail.



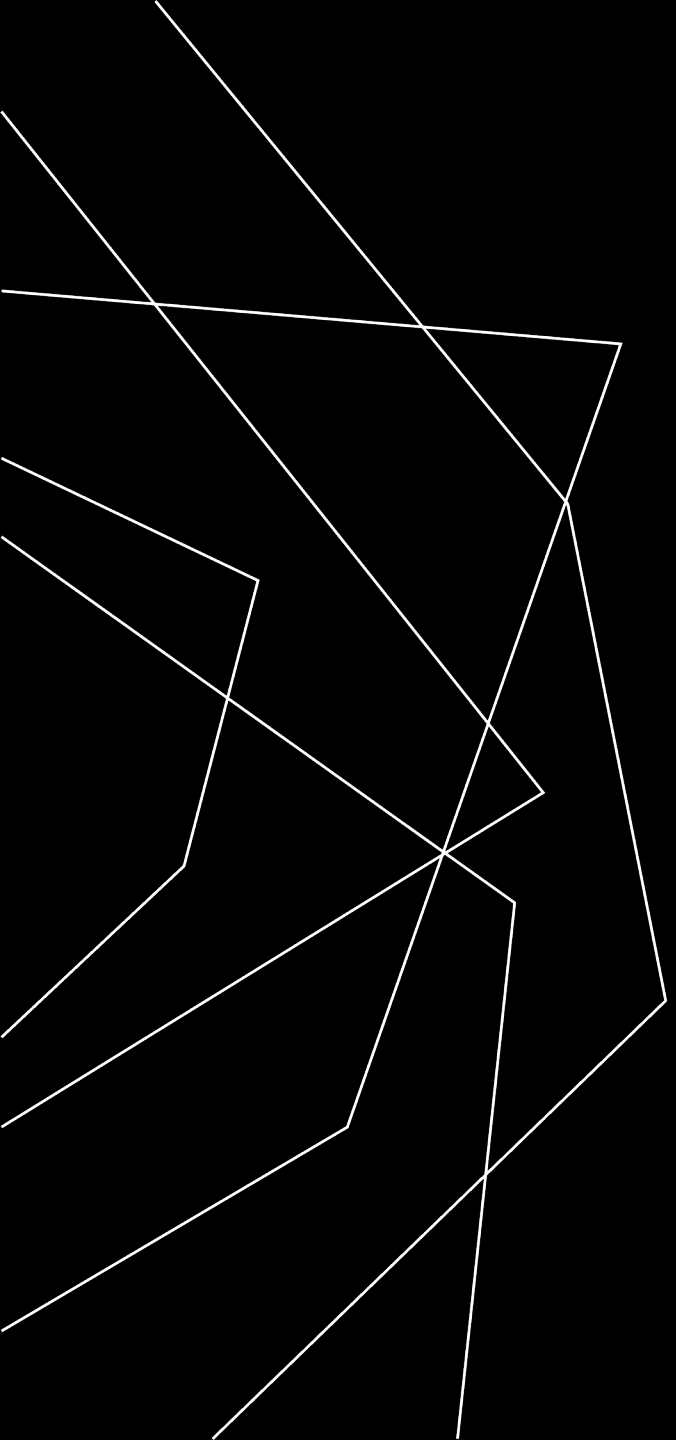
MAINTENANCE

- Software is subject to continual change, requiring the maintenance of an evolving system.
- Intuitive (N) individuals tend to be attracted to research and state-of-the-art development, while sensing (S) individuals tend to be more suited to maintaining and enhancing software systems due to their practical, realistic, and observant nature.
- Sensing (S) individuals prefer to perform tasks in a particular way based on past success, while intuitive (N) individuals prefer to perform tasks in a new and innovative way. Sensing individuals are well-suited to maintenance tasks due to their ability to focus on details and figure out how things work.
- Perceiver (P) individuals are more open to changes and adaptations and would be well-suited to maintenance tasks, particularly those that require problem-solving ability and a hands-on approach. Judger (J) individuals seek closure and may find maintenance tasks less satisfying.



SUBJECTIVE OPINION

- Software development involves a human element and is often a team effort.
- The success and failure of software projects are heavily influenced by the personality traits of those involved.
- There is no single personality type that is best suited for all aspects of software engineering.
- It is important to recognize and value the diversity of personality types in software engineering teams.
- Exposure to software psychology can help diversify software engineering teams and improve their effectiveness.
- Diversity of skills and personalities is necessary to solve the complex problems related to software development and maintenance.



THANK YOU