**1.1: Explain why professional software that is developed for a customer is not simply the programs that have been developed and delivered.**

**Customized:** Professional software development process is customer centric. They customize and tailor the software to meet the specific needs of each customer, and they are always open to feedback and making changes as needed.

**Use of agile development methodologies:** Agile development methodologies are a good fit for customer-centric software development because they are flexible and adaptable to change. This is important because the customer's needs may change over time, and the software must be able to change with them.

**Quality Assurance (QA):** Quality assurance (QA) is a systematic approach to ensuring that software meets its requirements and is free of defects. QA includes activities such as testing, reviews, and inspections.

**user experience (UX) design:** One more process the professional follow is User experience (UX) which is the process of creating software that is easy to use, enjoyable, and relevant to the user. This is essential for customer-centric software development, as the software must be easy for the customer to use in order to be successful.

"However, basic programs often prioritize output without considering user experience or potential real-time issues that could arise during production."

**Professional Software Process:** A paper with writing on it

Description automatically generated

**1.2: What is the most important difference between generic software product development and custom software development? What might this mean in practice for users of generic software products?**

The basic difference is, Generic software products are developed to meet the needs of a wide range of users, while custom software products are developed to meet the specific needs of a particular user or organization. **For example,** a basic calculator is a generic software product that can be used by anyone. However, if you need a calculator that can perform more complex operations, such as trigonometric functions or logarithms, then you would need to have a custom calculator developed.

**Cost:** Generic software products are typically less expensive than custom software products because the development costs are spread over a larger number of users. It won’t need to go through many UAT processes and QA process.

**Security:** Generic software products may not be as secure as custom software products because they are not designed to meet the specific security needs of a particular user or organization.

**Time:** The Generic software development takes very less time but whereas custom development take months to develop, test and deploy and sometimes it takes years if it is a huge application.

**1.3: Briefly discuss why it is usually cheaper in the long run to use software engineering methods and techniques for software systems.**

**Spend Quality of time to make sure no errors in the final product:**By investing time and effort upfront, potential issues can be identified and resolved early in the development process. This proactive approach prevents costly errors and the need for extensive rework later on.

**Enhanced productivity:** The adoption of software engineering methodologies and techniques can lead to increased efficiency in the software development process. Over time, this improved efficiency can result in cost savings by reducing the time and resources needed for software development.

**Better maintenance:** With time, all the processes in the application will be refined, resulting in a better maintenance process. This will help to ensure that no breakdowns occur, as the team will take into account feedback from previous breakdowns to prevent them from happening again.

**Better Documentation:** When we use software engineering methods and techniques, it makes a big difference in how well we document our software systems. This not only helps us understand and use the software more easily but also saves money in the long run.

**1.4: Software engineering is not only concerned with issues like system heterogeneity, business and social change, trust, and security, but also with ethical issues affecting the domain. Give some examples of ethical issues that have an impact on the software engineering domain.**

**Privacy Concerns:** Software engineers often deal with personal and sensitive data, and ethical concerns arise when this data is mishandled or misused. For example, collecting and storing user data without proper consent or security measures can breach privacy rights.

**Example:** Usually whenever developers stuck with some error, they simply post the error with ignoring sensitive data inside the screenshot which is misused by some people.

**Security:** Software engineers need to design secure software that protects against unauthorized access, modification, or destruction. This is especially important for software that is used in critical applications, such as healthcare or finance.

Software engineers need to use secure coding practices to protect their code from unauthorized access, modification, or destruction.

**Address Software Vulnerabilities:**

Another ethical concern in software engineering is vulnerabilities. It's our responsibility to find and fix these weaknesses in our software promptly. Neglecting to do so can put users at risk, possibly resulting in data breaches or other harmful activities. Ethical software development means taking proactive steps to reduce vulnerabilities.

Software engineers need to keep their software up to date with the latest security patches.

**1.5: Based on your own knowledge of some of the application types discussed in Section 1.1.2, explain, with examples, why different application types require specialized software engineering techniques to support their design and development.**

Mobile applications operate on a range of platforms, such as iOS and Android, and contend with resource limitations, including CPU and memory constraints. This necessitates the utilization of specialized techniques tailored to each platform, with a focus on optimizing performance and accommodating the diverse screen sizes of mobile devices.

**Example:** The WhatsApp app requires specialized techniques for creating a consistent user experience on both iOS and Android, efficient message sharing, and seamless integration with device cameras while making video calls.

**Game Development:**

Games demand real-time graphics rendering, physics simulations, and immersiveuser experiences. They require specialized techniques for game engines, 3D modeling, and optimization for various platforms.

**Example:** The development of a game like "Temple Run" requires specialized game engine technology, efficient rendering techniques, and multiplayer networking capabilities, and needs to work on different devices where they concentrate.

**Web applications** require specialized software engineering techniques to ensure that they are efficient, reliable, and secure.

**For example,** an e-commerce platform like Amazon requires techniques to handle thousands of users concurrently, process transactions securely, and ensure data privacy and loads as quickly as possible with no difference.

**1.6: Explain why the fundamental software engineering principles of process, dependability, requirements management, and reuse are relevant to all types of software systems.**

These principles bear significance across diverse software systems, contributing to the advancement of software development in aspects of quality, reliability, and efficiency. Adherence to these principles empowers software engineers to ensure that their software aligns with the needs of users and stakeholders while adhering to project timelines and budget constraints."

**Process:** A well-defined process acts as the cornerstone for developing software systems, be it for automobiles, medical devices, or financial trading platforms. Although the specific process may vary according to the system's nature, the core principles remain consistent."

**Dependability:** Dependability holds paramount importance in all software systems, with particular emphasis on safety-critical applications like medical devices and air traffic control systems. These systems require designs that prioritize superior reliability and fault tolerance."

**Requirements Management:** Effective requirements management stands as a crucial facet of software development, relevant to systems of varying complexities and involving diverse stakeholders. Careful collection and documentation of requirements ensure shared clarity among all project participants."

**Reuse:** The concept of reusability finds practical application in the development of software systems across a wide spectrum of uses. For instance, it facilitates the utilization of reusable components in constructing software for mobile devices, web applications, and embedded systems.

**1.7: Explain how electronic connectivity between various development teams can support software engineering activities.**

**Version control systems** (VCSs) like Git, Sub version (SVN) are commonly used by development teams to manage source code. These systems allow teams to track changes to code over time, and to revert to previous versions of the code if needed. VCSs also allow teams to collaborate on code, by allowing them to commit, merge, and synchronize code changes seamlessly. This ensures that everyone is working on the latest codebase and reduces the risk of conflicts.

**Knowledge Sharing (SharePoint):**

Electronic platforms can help teams share knowledge efficiently and effectively. Wikis, chat systems, and forums can be used to create and access documentation, best practices, and FAQs. This ensures that everyone on the team has access to the same information, and that newcomers can get up to speed quickly.

**Bug Tracking and Issue Management (Service Now / Jira):** Electronic systems enable teams to log and track software issues and bugs. This centralizes problem-solving efforts and ensures that no issues are overlooked, leading to more robust software.

**Communication and Standup collaboration Through Teams and Slack:**

Electronic connectivity enhances communication through email, chat, video conferencing, document sharing platforms, and collaborative tools like Microsoft SharePoint. Teams can maintain a clear and organized record of discussions, decisions, and project documentation using SharePoint's document management capabilities.