

UNIT 5

Ethical Decisions in Software Development and Ethics of IT Organizations

(8 Hours) — Cyber Law & Professional Ethics (CACS401)

***Please read:** These notes are prepared as a study aid and **may contain errors** or out-of-date details, especially section numbers, penalties, dates, case names and statistics. Always cross-check with your prescribed textbook, class lectures, and the official text of any law (Nepal Law Commission website) before relying on them in an exam or in practice.*

*Syllabus note: "H-IB Workers" is a typing error for **H-1B** visa workers. Immigration rules, fees and quotas change often (especially in the US), so treat the H-1B details as a general picture and check current official sources.*

5.1 Software Quality and Its Importance

Software quality is the degree to which a software product meets requirements, satisfies users' needs and works reliably, safely and securely. Defects (bugs) arise from unclear requirements, design mistakes, coding errors, poor testing, time pressure and weak management.

Quality attributes (based on ISO/IEC 25010): functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability and portability.

Why Quality Matters

- **Safety:** failures in medical, aviation, automotive and financial software can injure or kill people or cause huge losses.
- **Cost:** fixing defects after release costs far more than fixing them early (see Figure 5.1).
- **Reputation and trust:** users leave unreliable products.
- **Legal liability:** defective software can lead to claims for negligence, breach of warranty or contract, and product liability.
- **Ethical duty:** developers know risks that users cannot see; ethics requires that they act responsibly.

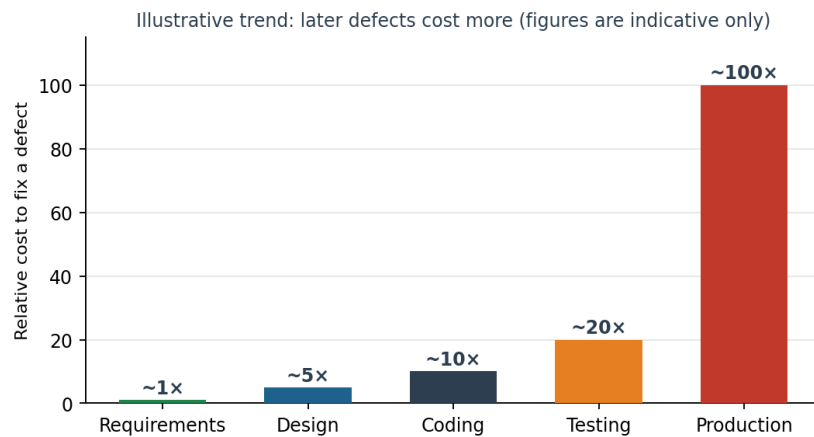


Figure 5.1: Relative cost of fixing a defect rises in later phases (illustrative)

Well-Known Software Failures

Case	What happened	Lesson
Therac-25 (1985 to 1987)	Radiation therapy machine gave massive overdoses due to software race conditions; several patients died or were seriously injured.	Safety-critical software needs rigorous testing and independent safeguards.
Ariane 5 (1996)	Rocket exploded seconds after launch because reused software caused a numeric overflow.	Reuse must be re-validated in the new context.
Knight Capital (2012)	Faulty deployment of trading software caused a loss of about 440 million US dollars in under an hour.	Deployment controls and testing are essential.
Boeing 737 MAX (2018 to 2019)	Flight-control software (MCAS) relying on a single sensor contributed to two fatal crashes.	Safety, transparency to users and regulators, and proper training.
CrowdStrike update (July 2024)	A faulty security-software update crashed millions of Windows computers worldwide.	Staged rollouts and thorough pre-release testing.

Note: Casualty figures and loss amounts vary between reports; the numbers above are approximate.

5.2 Strategies for Developing Quality Software

- **Clear requirements:** involve users and write testable requirements; manage changes carefully.
- **Sound development methodology:** plan, design, code, test and maintain using a suitable model (waterfall, iterative, Agile, DevOps). Safety-critical projects use more formal methods.
- **Maturity models:** the **Capability Maturity Model Integration (CMMI)** rates process maturity on levels: Initial, Managed, Defined, Quantitatively Managed and Optimizing. Higher levels mean more predictable quality. Standards such as ISO 9001 and ISO/IEC

12207 also guide processes.

- **Reviews and inspections:** peer code reviews and walkthroughs find defects early.
- **Testing at all levels** (see table below) and automated regression testing.
- **Static analysis, coding standards and secure coding** practices.
- **Configuration management and version control;** continuous integration and delivery.
- **Defect tracking and root-cause analysis** so the same mistakes are not repeated.
- **Realistic scheduling:** the "project triangle" (scope, time, cost) means squeezing time and cost hurts quality. Ethical managers should not force teams to ship known dangerous defects.
- **Post-release support:** patches, monitoring, incident handling and honest communication about known issues.

Test level	Purpose
Unit testing	Checks individual functions/modules (usually by developers).
Integration testing	Checks that modules work together.
System testing	Checks the complete system against requirements (functional, performance, security).
Acceptance testing	Users or customers confirm the product meets their needs (alpha/beta tests).
Regression testing	Re-tests after changes to be sure old features still work.

Black-box testing tests behaviour without looking at the code; **white-box testing** uses knowledge of the internal code. **Verification** asks "Are we building the product right?" (matches specification); **validation** asks "Are we building the right product?" (meets user needs).

Legal Issues in Software Quality

- **Warranty:** an express or implied promise that the product works as described. Vendors often add **disclaimers** and limits of liability in licences, but these cannot always excuse gross negligence or dangerous defects.
- **Negligence:** failing to use reasonable care in development. **Strict liability** can apply to defective products causing harm, although its application to pure software is still debated.
- **Fraud and misrepresentation:** knowingly hiding serious defects or overstating capabilities.

Example 5.1: Ship or Delay?

A team building a hospital medicine-dosage app finds intermittent calculation errors one week before release. The sponsor insists on the date because of a contract penalty. What should the team do?

Solution

Because patient safety is at stake, the errors must be treated as critical. The team should document the defect, quantify the risk, inform management and the client in writing, and recommend delay or a limited release with the faulty function disabled. The contract penalty is small compared with harm to patients and potential liability. If pressure continues, escalate and consider whistle-blowing (Section 5.6).

5.3 Use of Contingent Workers

Contingent workers are people who are not permanent, full-time employees of the organisation: temporary workers, part-timers, contract and freelance workers, consultants, and workers supplied by staffing agencies. Firms use them to meet peaks in demand, get special skills quickly, reduce costs (fewer benefits) and stay flexible.

Advantages for employer	Disadvantages / concerns
Flexibility to grow or shrink	Lower commitment; less loyalty and morale problems for regular staff
Access to specialised skills	Loss of knowledge when contract ends
Lower benefit and training costs	Risks to confidential data and intellectual property
Less long-term risk	Possible unfair treatment: low pay, no benefits, no job security

- **Legal issue:** misclassification, i.e., treating people who work like employees as "independent contractors" to avoid benefits and taxes. In *Vizcaino v. Microsoft* (US, 1990s to 2000), long-term "freelancers" were found to have been treated as employees and the company settled for about 97 million US dollars.
- **Co-employment:** when both the staffing agency and client control the worker, both may be legally responsible.
- **Good practice:** use contingent workers for genuinely temporary or specialised needs, give clear written contracts, pay fairly, protect confidential information with agreements, treat them respectfully, and do not use them to avoid legal duties.

- **Nepal:** the Labour Act, 2074 (2017) recognises categories such as regular, work-based, time-bound, casual and part-time employees, with specific rights.

5.4 H-1B Workers

The **H-1B** is a US non-immigrant work visa for foreign professionals in "specialty occupations" (such as IT, engineering, science, and finance) that require at least a bachelor's degree or equivalent. A US employer must sponsor the worker and file a **Labor Condition Application** promising to pay at least the prevailing wage. It is typically valid for three years and extendable to six. Demand exceeds the yearly numerical cap, so selection is done by lottery (a regular cap of 65,000 plus 20,000 for holders of US master's degrees).

Arguments in favour	Arguments against / ethical concerns
Fills genuine shortages of skilled workers	May displace or undercut local workers
Brings innovation and supports growth	Some outsourcing firms use the visa heavily to supply lower-cost labour
Gives skilled people global opportunity	Worker depends on employer for status, so may tolerate poor conditions
Helps companies remain competitive	Abuse: paying below the prevailing wage, "benching" workers

Reforms discussed include higher wage levels, wage-based selection, stricter enforcement and fees. In September 2025 the US government announced a large one-time payment (reported as 100,000 US dollars) for certain new H-1B petitions; its scope and legal status should be checked in current sources.

5.5 Outsourcing

Outsourcing means contracting an outside company to perform work that the organisation could do itself, such as software development, testing, help desks or payroll. **Offshore outsourcing** uses providers in other countries (e.g., IT firms in Nepal, India or the Philippines serving clients abroad); **nearshore** uses neighbouring countries; **onshore** uses local vendors.

Reasons to outsource	Risks and ethical issues
Lower labour cost and faster delivery	Job losses and morale impact at home
Focus on core business	Data security, privacy and IP risks across borders
Access to skills and 24-hour development	Poor communication, culture and time-zone differences

Reasons to outsource	Risks and ethical issues
Scalability and flexibility	Weak quality control, hidden costs, vendor lock-in
Creates jobs and income in developing countries (e.g., Nepal)	Poor working conditions or low pay at vendor

Strategies for successful outsourcing: choose vendors after due diligence; define scope and requirements clearly; sign a detailed contract and **service level agreement (SLA)** with quality metrics; include confidentiality, IP ownership and data-protection clauses; retain oversight and audit rights; plan for communication and cultural issues; treat vendor staff and displaced employees fairly.

5.6 Whistle-Blowing

Whistle-blowing is an effort to attract public attention to a negligent, illegal, unethical, abusive or dangerous act by a company or organisation. The whistle-blower is usually an insider who has evidence. **Internal** whistle-blowing reports to higher management or an ethics officer; **external** whistle-blowing reports to regulators, police, the media or the public.

Legal Protection

- **US:** False Claims Act ("qui tam" rewards for exposing fraud against the government), Whistleblower Protection Act (1989), Sarbanes-Oxley Act (2002) and Dodd-Frank Act (2010) protect against retaliation.
- **EU:** Whistleblower Protection Directive (2019).
- **Nepal:** protection is limited and scattered across laws (for example anti-corruption provisions); no comprehensive stand-alone whistle-blower act is widely cited. Verify the current position.

Guidelines for a Potential Whistle-Blower

- **Assess the seriousness** of the problem and be sure of the facts.
- **Begin documentation** (dates, documents, witnesses) while following confidentiality laws.
- **Attempt to address the issue internally** through supervisors, ethics office or hotline.
- **Consider the risks:** job, career, legal action, personal life.
- **Get legal advice** before going outside.
- **Then go external** in an appropriate way (regulator first is often safer than the media).

Organisations should create safe, anonymous reporting channels, investigate complaints seriously, and forbid retaliation. **Examples:** Frances Haugen (Facebook internal documents,

2021), Edward Snowden (NSA surveillance, 2013, a highly debated case), and the Boeing 737 MAX safety concerns raised by employees. Whether a disclosure is ethical depends on the truth, public interest, proportion of harm and whether internal channels were tried.

Example 5.2: Should She Blow the Whistle?

Anita, a QA engineer at a payment-app company, finds that customer card numbers are stored unencrypted. Her manager says the fix is "not a priority" and asks her to keep quiet.

Solution

Follow the steps: (1) confirm the facts; (2) document; (3) report to the security officer, compliance team or senior management; (4) if ignored, seek legal advice and report to the regulator (e.g., central bank/cyber authorities); (5) avoid public leaks of actual card data. The duty to prevent serious harm to customers outweighs loyalty to the manager, and following the escalation path protects her legally and ethically.

5.7 Green Computing

Green computing (green IT) is the environmentally responsible design, manufacture, use and disposal of computers, servers and related systems to reduce energy use, carbon emissions and waste. IT consumes large amounts of electricity (especially data centres) and creates **e-waste** that contains hazardous materials such as lead, mercury and cadmium if disposed of improperly.

Area	Green practice
Energy use	Energy-efficient hardware (Energy Star labels, EPEAT rating), power-saving modes, turning off idle devices, efficient cooling; measuring data-centre efficiency with PUE (power usage effectiveness).
Virtualisation and cloud	Fewer physical servers, higher utilisation, shared efficient data centres.
Renewable energy	Use solar, hydro or other renewable power for offices and data centres.
Paper and travel	Paperless workflows; video conferencing and remote work to cut travel.
Product life cycle	Longer device life, repair and upgrade, refurbishment and reuse.
E-waste	Recycle through authorised handlers; extended producer responsibility (manufacturers take back devices); avoid dumping.
Green software	Efficient algorithms and code that use less computing power.

- **Regulations and standards:** EU WEEE and RoHS directives (e-waste recycling and restriction of hazardous substances), the Basel Convention (1989) on transboundary movement of hazardous waste, ISO 14001 environmental management.

- **Nepal:** solid-waste rules exist (Solid Waste Management Act, 2068), but e-waste handling is still developing; many devices are imported second-hand. Verify current regulations.
- **Benefits:** lower costs, lower carbon footprint, better brand image and compliance.
Challenges: upfront cost, lack of awareness and weak recycling infrastructure.

5.8 Important Short Questions (Exam Practice)

- What is software quality? Why is it important? Give examples of software failures.
- Describe strategies for developing quality software. What is CMMI?
- Who are contingent workers? Discuss advantages, disadvantages and ethical issues.
- What is an H-1B visa? Discuss the ethical concerns.
- Explain outsourcing, its benefits and risks. How can it be managed ethically?
- What is whistle-blowing? Explain the steps a whistle-blower should follow.
- What is green computing? List green IT practices.

Key Takeaways

- ✓ Quality software protects safety, saves cost, builds trust and reduces legal liability.
- ✓ Strategies: clear requirements, good process (CMMI), reviews, layered testing, defect tracking and ethical scheduling.
- ✓ Contingent workers give flexibility, but avoid misclassification and protect their rights and your IP.
- ✓ H-1B: skilled-worker visa; debate centres on shortages versus wage undercutting.
- ✓ Outsourcing needs due diligence, SLAs, security and fair treatment.
- ✓ Whistle-blowing: verify, document, try internal channels, get legal advice, then go external.
- ✓ Green computing reduces energy use and e-waste through efficient, reusable and recyclable IT.

— End of Unit 5 —