Doing Projects

We Solve Problems by Managing Complexity

Project Related Rules

Rumsfeld's Law

> We don't know what we don't know (Unknown unknowns) until we know what we don't know (Known Unknowns).

1. **Start Early:** Begin quickly to reveal what you don’t know.
2. **Prototype First:** Test feasibility before the MVP to ensure you’re solving the right problem.
3. **Seek Clarity:** Work hard to gain a clear vision as a problem solver.

Effective first, efficient later

> Planning the fastest trip to New York is useless if you really need to be in Los Angeles.

1. **Focus on Impact:** Solve the right problem before worrying about speed or cost.
2. **Validate Direction:** Early solutions may be rough, but they prove value and feasibility.
3. **Refine for Efficiency:** Once effectiveness is clear, streamline and optimize.
4. **Avoid Premature Optimization:** The first goal is to ship the MVP, not perfect it.

Make it a Game, Enjoy Small Victories

> Big achievements are just a collection of small wins stacked together.

1. **Gamify the Work:** Treat challenges like levels to beat.
2. **Celebrate Progress:** Every small win builds momentum.
3. **Stay Motivated:** Fun fuels persistence through obstacles.
4. **Use All Course Materials:** Treat homework, exams, and quizzes as opportunities to practice problem-solving, not burdens to avoid.

Make it Work, then Make it Better

> A rough solution that works is more valuable than a perfect one that never ships.

1. **Start Simple:** Deliver something functional first.
2. **Prove Value:** Show it solves the problem before refining.
3. **Iterate Later:** Improve design, speed, and polish after validation.
4. **Avoid Perfection Trap:** Progress beats endless tweaking.

Think in Systems, Build the System

> Strong systems outlast quick fixes and scale beyond individual effort.

1. **Vertical Slice First:** Build the system to deliver end-to-end value instead of isolated layers.
2. **Sustainable Foundations:** Design systems and processes with growth and change in mind.
3. **Think Solutions in Systems:** Let solution patterns emerge naturally within the system.
4. **System as Accelerator:** Once built, the system makes every next project faster and easier.

System-Driven Processes

> A good system creates processes that run and improve themselves.

1. **Self-Improving:** The process continuously makes the process better.
2. **Reduce Mental Burden:** Free people from micromanagement and memory load.
3. **Consistency:** Ensure reliable results regardless of who executes the task.
4. **Scalability:** Allow teams to grow without adding complexity.

Roles and Responsibilites

Professor (as Manager)

- **Manage People & Politics:** Navigate complexity and team dynamics.
- **Report Upward:** Communicate with high-level officials.
- **Evaluate Teams:** Assess tech leads and team members.
- **Decide Careers:** Handle promotions, raises, hiring, and firing.
- **Support Problem-Solving:** Provide guidance and resources to resolve issues.

Team Leaders (Tech Leads)

- **Manage Complexity:** Solve problems by organizing people and tasks.
- **Report to Managers:** Keep leadership informed.
- **Set Direction:** Define goals and schedules.
- **Drive Progress:** Ensure team members move forward.
- **Communicate Upward:** Report team progress to managers.

Team Members (Senior Software Engineers)

- **Tackle Complexity:** Solve problems through design and implementation.
- **Design Architecture:** Create modules and interfaces.
- **Build & Test:** Write code and tests when needed.
- **Raise Issues:** Signal problems early to get them resolved.
- **Mentor Others:** Support junior developers and interns.

Team Rules

No Surprises

> Unexpected problems break trust more than difficult problems.

1. **Communicate Early:** Raise issues as soon as they appear.
2. **Be Transparent:** Share progress, blockers, and risks openly.
3. **Set Expectations:** Keep managers and teammates informed.
4. **Avoid Last-Minute Shocks:** No one should be blindsided.

Show No Emotions

> Professionalism means staying calm, even under pressure.

1. **Stay Calm:** Don’t let frustration or excitement cloud communication.
2. **Focus on Facts:** Discuss data and solutions, not feelings.
3. **Maintain Composure:** Handle conflicts without visible anger or stress.
4. **Build Trust:** A steady presence reassures the team and clients.

Be a Professional

> Professionalism is about actions and results, not titles.

1. **Deliver on Promises:** Do what you say, on time.
2. **Respect Others:** Treat everyone with fairness and courtesy.
3. **Take Responsibility:** Own mistakes and learn from them.
4. **Strive for Quality:** Aim for excellence in every task.

Understand Others before Being Understood

> Listening deeply builds trust faster than speaking first.

1. **Seek Context:** Learn the origin of the problem and who's involved.
2. **Listen Actively:** Let others feel heard before offering input.
3. **Empathize First:** Acknowledge concerns and perspectives.
4. **Respond Thoughtfully:** Solutions land better when people feel understood.

Evaluation

Based on Rubrics

- Rubrics check whether you deliver on your promises according to plan.
- Evaluations are based directly on these rubrics.
- Focus on progress, clear communication, solid design, and flexible systems—then you'll earn the points and trust you deserve.

Three Options on Using Generative AI

> Generative AI is like the invention of fire or electricity—we must use it, but we should use it wisely.

1. **No AI Use:** Focus on strengthening raw coding skills without assistance.
2. **AI as Support:** Leverage AI for tests, refactoring, and code improvements to accelerate progress.
3. **Vibe Coding:** Collaborate with AI to produce impressive results that go beyond what you could achieve alone.

Low Quality Results will Earn Less

> As professional software engineers, low-quality results are not acceptable.

1. **Individual Responsibility:** Teammates may support and encourage, but each member is accountable for their own results.
2. **No Progress Means Low Quality:** Lack of progress damages the team just as much as poor output.
3. **Objective Evaluation:** Results will be judged professionally by peers and leaders—without emotions.

Last Minute Sprints Will Earn Less

> Wasting time and rushing at the last moment is the most dangerous habit for professionals.

1. **Plan Wisely:** Schedule your time to prevent last-minute crunches.
2. **Communicate Early:** Share issues immediately with teammates, tech leads, and managers.
3. **No Compensation:** Fancy results cannot make up for last-minute work—trust may already be broken.
4. **Respect Deadlines:** Treat the **dead**line as final—set your own target at least one week earlier.

Project Results Are Your Portfolio

> Add at least two projects to your portfolio after completing an ASE course.

1. **Prove Capability:** Show real problem-solving skills.
2. **Stand Out:** Only résumés with portfolios get noticed—keep a one-page version ready.
3. **Showcase & Promote:** Use GitHub as your hub and highlight projects on LinkedIn.
4. **Be Proud:** Share both victories and how you overcame failures—your future peers will want to hear about them.

The Format of Project Progress (Team/Individual Project)

Overall Process (Team)

1. **Progress Pages:** Canvas pages are created for every member and leader for the team project.
2. **Set Plans:** Each leader and member defines goals, deadlines, and milestones (HW2).
3. **Update Regularly:** Use the pages to track and update goals and progress on a weekly basis.
4. **Weekly Presentations:** Team leaders review and discuss progress at the first meeting each week.

Weekly Presentations (Team)

1. **Leaders Monitor Only:** Team leaders don’t ask for submissions—just check progress pages.
2. **No Public Call-Outs:** If a member makes no progress, leaders don’t discuss it in presentations.
3. **Know Your Role:** Non-technical team issues are for managers, not tech leads, to resolve.
4. **Objective Evaluation:** Leaders and members assess each person fairly and without emotion.

Celebration Moment

> We will celebrate these milestones, each followed by a presentation.

1. **Project Kickoff:** Teams set goals, deadlines, and milestones, and upload them to start.
2. **Prototype Demo:** Celebrate when the prototype is built and presented (if required).
3. **MVP Launch:** Mark the moment when the minimum viable product is ready.
4. **Project Completion:** Final celebration when the project is finished.

Overall Process (Individual Project)

1. **Progress Pages:** Canvas pages are created for every member for the individual project.
2. **Set Plans:** Each leader and member defines goals, deadlines, and milestones (HW2).
3. **Update Regularly:** Use the pages to track and update goals and progress on a weekly basis.
4. **Presentations:** Each team member presents their project in class; if time is limited, presentations may be shortened.

Presentation

Know your audience

> Explaining everything in technical jargon is like speaking a foreign language to someone who just wants the story.

1. **Identify Stakeholders:** CEOs, managers, product owners, sales, and real clients all have different needs.
2. **Tailor Communication:** Adjust depth—big picture for executives, details for developers.
3. **Focus on Value:** Show how your work impacts business goals, not just technical wins.
4. **Build Trust:** When people feel understood, they'll trust your solutions more.

In Your Presentation

> You are an actor—stay natural no matter what happens.

1. **Irregularities Stand Out:** The audience remembers what breaks the flow.
2. **Audience First:** Say what they need to hear, not just what you want to share.
3. **Be Concise:** Keep it short and to the point.
4. **Add Page Numbers:** Always number your slides for clarity and reference.

My Recommendations on Projects

1. **SEFE First:** Always remember the first rule—*Start Early to Finish Early*.
2. **No Surprises:** Keep others updated on your progress.
3. **Control What You Can:** Don’t waste energy on what’s beyond your control—set goals, deadlines, milestones, and embrace small wins and failures.
4. **Learn & Celebrate:** Enjoy your successes and learn from your failures.
5. **Growth Over Points:** Focus on developing skills, not just earning grades.
6. **Ask for Help:** I’ll support you—just let me know when you need it.