Technical Risk and Technical DebtScrum Teams always need to balance the need for new user features with keeping the product technically sound.

This often means assessing **technical risks** and prioritising between satisfying user requests and fixing **technical debt**—shortcuts or incomplete work that can cause problems over time.

What is Technical Debt?

Technical debt happens when technical maintenance is delayed and builds up over time. This can be due to a priority call for things that the user / business desires to be delivered first. Also it can be influenced by the cost of doing the work at the point in time it is discovered. Examples include temporarily patching bugs instead of fixing the root cause or skipping software updates due to time constraints. To manage this, Scrum Teams may add backlog items to address technical debt bit by bit. However these may still get delayed and become technical debt.

Balancing User Requests and Technical Debt

- **User Requests**: These bring new features, boost user satisfaction, and help grow the product.
- **Technical Debt**: If ignored, it slows development, increases errors, and limits the product's ability to grow.

How to Prioritize To Avoid Technical Risk

1. **Align with Product Goals**: Focus on how fixing the tech issue will help the long-term vision.
2. **Assess Impact**: Look at the value of new features vs. risks of technical debt.
3. **Listen to Users**: Prioritize features users want most and how potential tech issues will help them.
4. **Review Technical Health**: Regularly check and fix areas where debt is causing issues.
5. **Reduce Debt Gradually**: Dedicate part of each Sprint to clearing debt over time.
6. **Balance the Backlog**: Mix user features with technical fixes to make steady progress on both fronts.

Communicate Clearly

Be open with stakeholders about how priorities are set. When everyone understands the reasoning, it's easier to get support for balancing user needs and technical fixes.

Conclusion

Scrum Teams must carefully juggle short-term user demands and long-term product & technical health. By focusing on goals, assessing impact, and staying transparent, they can create a product that satisfies users today while remaining strong for the future.