**Module 11: Requirements Decomposition**

1. **Importance of Requirements Decomposition**
   * Decomposition is critical in **large and critical systems** like military, aerospace, and telecom​(Module 11 - Requirement…).
   * Software engineering may become more automated, but **requirements decomposition** remains a human-centric task​(Module 11 - Requirement…).
2. **Types of Requirements Decomposition**
   * **Flow-Down**: Assigning requirements to appropriate subsystems​(Module 11 - Requirement…).
   * **Refinement**: Adding details and constraints to make requirements actionable by design and implementation teams​(Module 11 - Requirement…).
   * **Completion**: Ensuring all requirements are fully traced back to the code (used in critical industries like aerospace)​(Module 11 - Requirement…).
3. **Examples of Decomposition**
   * **Crew Alerting System (CAS)**: System requirements assigned to subsystems like UI or Information Services​(Module 11 - Requirement…).
   * **Resource Allocation**: Example from telecom where time budgets are allocated to different subsystems to fulfill performance requirements​(Module 11 - Requirement…).
4. **Key Challenges**
   * **Derived Requirements**: Can either mean requirements traced from a higher level or added requirements not traced back to a specific user need​(Module 11 - Requirement…).
   * **Terminology Confusion**: Different industries (e.g., aerospace vs. military) use different definitions for the same terms like "derived requirements"​(Module 11 - Requirement…).
5. **Why Decomposition Matters**
   * Decomposition provides a comprehensive trace between requirements and system components, ensuring everything is accounted for during delivery, maintenance, and evolution​(Module 11 - Requirement…).

**Module 12: Requirements Management**

1. **Key Areas of Requirements Management**
   * **Traceability**: Ensuring requirements can be traced back to their origins and forward to design, implementation, and testing​(Module 12 - Requirement…).
   * **Planning for Change**: Change is inevitable, and requirements management must account for new and changing needs​(Module 12 - Requirement…).
   * **Methodology**: Using a structured approach to manage requirements through their lifecycle​(Module 12 - Requirement…).
   * **Tools**: CASE tools are crucial for managing complexity​(Module 12 - Requirement…).
2. **Managing Change**
   * **Change as Risk**: Requirements can change due to evolving user needs, business environments, or unexpected discoveries​(Module 12 - Requirement…).
   * **Five Types of Change** (from Harker et al.):
     + **Mutable**: Changes in the customer's business environment.
     + **Emergent**: New requirements that evolve as understanding improves.
     + **Consequential**: Changes requested after users see the system.
     + **Adaptive**: Changes after users find better ways to use the system.
     + **Migration**: Supporting current users during rollout​(Module 12 - Requirement…).
3. **Requirements Management Maturity Model**
   * **Level 0 (Chaos)**: No requirements management leads to poor quality and missing functionality​(Module 12 - Requirement…).
   * **Level 1 (Written)**: Requirements are documented, forming a contract with customers and the implementation team​(Module 12 - Requirement…).
   * **Level 2 (Organized)**: Requirements are identified, persisted, and versioned​(Module 12 - Requirement…).
   * **Level 3 (Structured)**: Classify requirements, track dependencies, and state priorities​(Module 12 - Requirement…).
   * **Level 4 (Traced)**: Ensure both upward and downward traceability of requirements​(Module 12 - Requirement…).
   * **Level 5 (Integrated)**: Fully integrated with the overall project process; no changes happen without a full review​(Module 12 - Requirement…).
4. **Change Request Management (CRM)**
   * **Centralized Change Control Board (CCB)**: Serves as the authority for approving changes, managing risks, and balancing competing interests​(Module 12 - Requirement…).
   * **Process**: Capture change requests, evaluate the broader impact, and ensure controlled modifications​(Module 12 - Requirement…).
5. **Traceability**
   * Ensures **quality**, **impact analysis**, and **verification** by linking requirements to related artifacts like design, tests, and code​(Module 12 - Requirement…).
   * Traceability is essential for managing complex projects where changes in one area can have cascading effects​(Module 12 - Requirement…).